Image Cover Sheet

CLASSIFICATION	SYSTEM NUMBER
UNCLASSIFIED	509553

TITLE

ANALYSIS OF PARTICULATE COMPOSITE BEHAVIOR BASED ON NONLINEAR ELASTICITY AND AN IMPROVED MORI-TANAKA THEORY

System Number:

Patron Number:

Requester:

Notes:

DSIS Use only:

Deliver to:

UNCLASSIFIED

DEFENCE RESEARCH ESTABLISHMENT
CENTRE DE RECHERCHES POUR LA DÉFENSE
VALCARTIER, QUÉBEC

DREV – R – 9815

ANALYSIS OF PARTICULATE COMPOSITE BEHAVIOR BASED ON NONLINEAR ELASTICITY AND AN IMPROVED MORI-TANAKA THEORY

by

F. C. Wong

September/septembre 1998

Approved by/approuvé

Section Head/Chef de section

15 October 98

Date

SANS CLASSIFICATION

ABSTRACT

A micromechanical model for the analysis of particulate mechanical behavior is presented. Nonlinear effects are introduced in the model by a nonlinear elastic description of the matrix and through a modulus degradation routine. The first part of the study uses the experimental data from a range of glass bead/HTPB composites to back calculate the model parameters. The results showed that the model gave a good representation of the processes believed to control mechanical behavior. These processes include partial particle debonding and progressive debonding from the largest to smallest particles throughout the strain history. The second part of the study examines the sensitivity of the model results to small changes in the adjustable input parameters. The residual bond in a debonded particle was found to have a dominating effect on the calculated results. Based on the sensitivity results, "best guess" interaction and debonding parameters were selected to examine the predictive capability of the model. For glass bead/HTPB composites, the predicted composite stresses were within 10% of the experimental data. Dilatation was usually over-predicted. For glass bead/polyethylene and glass bead/polyurethane data found in the literature, predicted composite stresses were within 15% to 24%, respectively. The results showed that the model was capable of predicting the mechanical behavior of composites comprised of glass beads in HTPB, PU or HDPE matrices as long as characteristic adhesive parameters were available for each system.

RÉSUMÉ

Un modèle micromécanique pour l'analyse du comportement mécanique de composites chargés est présenté. Les effets non-linéaires sont introduits dans le modèle à l'aide d'une description élastique non-linéaire de la matrice et d'un algorithme de dégradation du module. Dans la première partie de l'étude, les paramètres du modèle sont estimés à l'aide de données expérimentales obtenues sur une gamme de composites constitués de billes de verre et de polybutadiène à terminaisons hydroxyles (PBHT). Les résultats ont montré que le modèle donne une bonne représentation des processus qui contrôlent le comportement du composite. Ces processus incluent le décollement partiel au niveau de l'interface particules/matrice et le décollement progressif des plus grandes aux plus petites particules tout au long du processus de déformation. Dans la deuxième partie de l'étude, la sensibilité du modèle aux petites variations des paramètres estimés est examinée. L'adhésion résiduelle

de particules partiellement libérées de la matrice s'est révélée avoir un effet dominant dans la solution. Avec ces résultats, les paramètres estimés pour les facteurs d'interaction et de décollement ont été choisis pour étudier la capacité du modèle à prédire le comportement mécanique de certains composites. Pour les composites billes de verre/PBHT, une différence de 10% entre les prédictions et les données expérimentales a été observée, la dilatation étant habituellement surestimée. Les données tirées de la littérature pour des composites billes de verre/polyéthylène haute densité (PEHD) et billes de verre/polyuréthanne (PU) sont prédites dans une marge de 15% et 24% d'erreur respectivement. Cette étude a démontré qu'il est possible de prédire adéquatement le comportement mécanique des composites à base de billes de verre dans des matrices PBHT, PU ou PEHD à condition que les valeurs caractéristiques du paramètre d'adhésion soient disponibles pour chaque système.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Propellants are presently characterized from a macroscopic point of view. This means that the mechanisms that govern material behavior are lumped together and measured as a unit to produce a material property. This approach does not provide the quantitative information required for modifying a formulation. To meet this need, an analytical model that predicts the material properties from knowledge of factors such as particle size distribution, volume fraction of particles, adhesion energy and polymer properties is required. This ability to predict mechanical properties has important consequences for the determination of rocket motor service life. If the properties of the motor grain can be predicted before the propellant is cast, motor service life can be determined. If the calculated service life is deemed too short, the model can be used to guide the type of adjustments that need to be made to extend the service life of the motor. This capability would represent major savings in development and life cycle management costs because service life related problems could be resolved before the motor is fielded.

In recent years, researchers in the propellant industry have begun to use composite materials concepts for predicting the stress-strain behavior of propellants. These concepts, based on a microscopic point of view, take into account the size, shape and quantity of filler introduced into polymeric matrices. Previously, the merits of a micromechanical model based on an improved Mori-Tanaka (M-T) method was evaluated. Results showed that at high inclusion volume fractions, correct modulus predictions could only be made by accounting for particle interaction effects. The performance of this model was limited by the assumed linear elastic matrix.

In this report, a new implementation of the M-T micromechanical model that accounts for a nonlinear elastic matrix was developed. The first part of the study used experimental data from a range of glass bead/polybutadiene composites to back calculate the model parameters. The results showed that the model gave a good representation of the processes believed to control mechanical behavior. The second part of the study examined the sensitivity of the model results to changes in the input parameters. The residual bond in the debonded particles was found to have a dominating effect. Based on the sensitivity analyses, "best-guess" parameters were selected to examine the predictive capability of the model for a variety of particulate composites. The results showed that the model was capable of predicting the mechanical behavior as long as suitable values for critical stress and adhesion energy were available. Thus, it is possible to use this analytical model to address formulation problems.

NOMENCLATURE

square brackets [ ] denote dimensions of the variable

A	-	debonded surface area of inclusion, $[L^2]$
A_{ijkl}	=	$(C^i_{ijmn} - C^o_{ijmn})^{-1} \cdot C^o_{mnkl}$, $[FL^{-2}]$
a_i	-	fitting coefficients for matrix polynomial, [–]
B_{ijkl}	=	$(C^v_{ijmn} - C^o_{ijmn})^{-1} \cdot C^o_{mnkl}$, $[FL^{-2}]$
$\overline{C}_{ijkl}$	-	average elastic constants of composite, $[FL^{-2}]$
C^o_{ijkl}	-	elastic constants of comparison material, $[FL^{-2}]$
C^r_{ijkl}	-	elastic constants of phase-r material, $[FL^{-2}]$
c^i	-	volume fraction of inclusions, [–]
c^i_o	-	initial volume fraction of inclusions, [–]
c^r	-	volume fraction of phase-r inclusion, [–]
c^v	-	volume fraction of voids or vacuoles, [–]
$d\mathcal{E}_{area}$	-	energy dissipated through surface creation, $[FL]$
$d\mathcal{E}_{mod}$	-	net internal strain energy due to modulus degradation, $[FL]$
E_c	-	average composite tensile modulus, $[FL^{-2}]$
E_i	-	inclusion tensile modulus, $[FL^{-2}]$
E_{ij}	-	tensile modulus in ij-direction, $[FL^{-2}]$
E_o	-	matrix tensile modulus, $[FL^{-2}]$
E^r_o	-	reference stress relaxation modulus at $t = 1$, $[FL^{-2}]$
$E(t)$	-	stress relaxation modulus, $[FL^{-2}]$
F_b	-	partial debonding factor, [–]
F^{err}_b	-	average absolute error for finding F_b, [–]
G_c	-	adhesion energy, $[FL]$
G^{app}_c	-	apparent adhesion energy, $[FL]$
G_o	-	matrix shear modulus, $[FL^{-2}]$
I_{ijkl}	-	identity matrix, [–]
K_o	-	matrix bulk modulus, $[FL^{-2}]$
M_ϵ	-	composite strain magnification factor, [–]
m	-	exponent in stress relaxation power law, [–]
n	-	log standard deviation in particle distribution, [–]
r	-	particle radius, $[L]$
$\overline{r}$	-	mean particle radius, $[L]$
S^r_{ijlk}	-	Eshelby matrix of phase-r material, [–]
t	-	time, $[T]$
dV/V	-	composite volume dilatation, [–]
V_o	-	specimen volume, $[L^3]$
Y	-	interaction function, [–]
Y_m	-	interaction function multiplier, [–]
Y^{err}_m	-	average relative error for finding Y_m, [–]

ϵ_{cr} - critical strain, $[L/L]$
ϵ_{ij} - average composite strain in ij-direction, $[L/L]$
ϵ_c - average composite uniaxial strain, $[L/L]$
ϵ_o - average uniaxial matrix strain, $[L/L]$
Γ^r_{ijkl} - correction matrix of phase-r material, $[-]$
ν_{ij} - Poisson ratio in ij-direction, $[-]$
ν_o - matrix Poisson ratio, $[-]$
σ_{cr} - critical stress, $[FL^{-2}]$
σ_{ij} - composite stress in ij-direction, $[FL^{-2}]$
σ_c^{eng} - average composite engineering stress, $[FL^{-2}]$

σ_c^T - average composite true stress, $[FL^{-2}]$

1.0 INTRODUCTION

The prediction of solid propellant mechanical behavior has attracted the attention of many researchers over the years. During that time, much of the effort was concentrated on the development of phenomenological models that treated propellant as a homogeneous material. Other researchers took a different approach and treated propellant as a particulate composite. The micromechanical analyses carried out by these researchers provided insight to the physical processes that controlled particulate composite or propellant behavior.

It was recognized early on that volume change was closely linked to the nonlinear behavior a particulate composite exhibited as it was loaded. In Refs. 1 and 2, the authors hypothesized that particulate composites exhibited three distinct regions in their stress-strain behavior. The initial region was controlled primarily by the initial inclusion concentration and the matrix properties. A transition region delimited the beginning and completion of inclusion debonding. Application of strain in the last region was believed only to stretch the binder and enlarge existing vacuoles. A vacuole was defined as a spheroidal air pocket which entrapped a debonded inclusion.

A different hypothesis on inclusion debonding was advanced in Refs. 3 and 4. Here the authors assumed that inclusion debonding was continuous throughout the strain history. Constituents were assumed to be linear elastic. Their micromechanical model parameterized several experimentally observed phenomenon such as the relationships between reinforcement and particle size (Refs. 5 - 9), reinforcement and adhesion (Refs. 10 - 13) and inclusion concentration and modulus (Refs. 14, 15) in a computationally tractable manner. In the range of analysis techniques covering semi-empirical formulations (Refs. 16, 17), variational formulations (Refs. 18 - 20) and approximate methods (Refs. 21 - 24), their model fell

in the approximate category because debonding at a microscopic level was quantified by modulus prediction routines that calculated average composite properties.

An evaluation of the combined concentration decrease/void addition model in Refs. 3 and 4 was made in Ref. 25 using a glass bead/polyethylene composite. The model decreased the inclusion concentration as inclusions were debonded and replaced them by equivalent sized voids. A void was defined as a spherical air pocket with isotropic properties. Following Refs. 3 and 4, the linear elastic constituent assumption was retained. It was concluded that the model could predict the mechanical behavior of highly loaded composites if a representative adhesion energy was available and if matrix nonlinearity was accounted for.

In Refs. 26 and 27, the model in Ref. 25 was improved by implementing a modulus prediction routine based on the Mori-Tanaka (M-T) method (Ref. 28) and the work of Ju and Chen (Ref. 29). The addition of Ju and Chen's modulus correction matrix to the M-T method accounted for additional reinforcing effects due to particle interaction. The improved M-T routine allowed debonded inclusions to be modeled as vacuoles by attributing orthotropic elastic constants to them.

The merits of the improved M-T routine were evaluated using literature data. Results showed that at high inclusion volume fractions, correct modulus predictions could only be made by accounting for particle interaction effects. Comparison of the new micromechanical model based on the improved M-T method with experimental data showed that modeling debonded particles by vacuoles instead of voids gave more representative results. The performance of the new model was limited like the original model by the assumed linear elastic matrix.

In this document, the implementation of a routine in the M-T based micromechanical model to account for a nonlinear elastic matrix will be described. The validity of the assumptions on which the model is based will then be explored. This will show that the assumption of continuous inclusion debonding throughout the strain history is justifiable. A comparison of calculated and experimentally measured parameters that are believed to control composite behavior will be made using the results from glass bead/polybutadiene, glass bead/HDPE and glass bead/polyurethane composite systems. A sensitivity analysis will show that the micromechanical model can produce accurate results as long as suitable values for critical stress and adhesion energy are available. This work was performed under TNS 03ee15 Service Life Prediction between July 1995 and December 1996.

2.0 BACKGROUND

In 1963, Freudenthal and Shinozuka (Ref. 30) examined the shrinkage stresses and strains in a viscoelastic thick-walled cylinder of infinite length. The main effort of the work was focused on deriving the analytical equations to predict the cylinder stress state. In order to pose a tractable problem, the material behavior was idealized by a standard viscoelastic solid.

Schapery (Ref. 31), in 1968, developed an approximate method to carry out a structural analysis of a long, circular port grain under transient and ignition pressurization. Within a thermodynamic framework that related the finite strains to a set of generalized forces through a virtual work condition, propellant shear properties were accounted for using a linear viscoelastic material model. Nonlinear behavior was introduced through a series of functions that modified the constitutive equation.

Farris (Ref. 32) developed a viscoelastic constitutive model to analyze stresses during unloading and reloading conditions. By tracking previously experienced maximum strains in the strain history through the use of Lebesque norms, he showed that this measure could be used to represent microstructural damage such as vacuole dilatation.

Lee (Ref. 33) and Cost (Ref. 34) extended the commonly used linear viscoelastic theory to deal with experimentally observed coupled straining-cooling behavior. Lee introduced an exponential function for reduced time along with a thermal relaxation function in his constitutive equation. Cost used a power law form of reduced time. Additional transient thermal tests were needed to characterize the constants found in these new models.

In 1983, Swanson and Christensen (Ref. 35) proposed a model based on large-strain stress and strain tensors to handle geometric nonlinearities. Material nonlinearities such as coupled strain-cooling behavior was handled with a strain softening function. Strain rate related nonlinearities were included through a second function inside the Duhamel integral. The authors stated that the constants in phenomenological formulations could sometimes be used to deduce micromechanical mechanisms.

Buckley proposed an alternative method for calculating the stresses in a linear viscoelastic solid as it was simultaneously strained and cooled (Ref. 36). Using a generalized Maxwell model and simplifying it to a thermoelastic form by cooling the material under a special sequence, he found that it was possible to eliminate the dependency on time. This method was useful as long as the cooling rate was slow enough to meet his time criterion.

In Ref. 37, Burke proposed a constitutive model based on an extended Mooney-Rivlin strain energy function. Material behavior was described through a fourth-order stress relaxation tensor. The model formulation required that two sets of material constants be

available so an ancillary program was written to automate calculating the constants.

Özüpek (Ref. 38) applied Swanson's ideas (Ref. 35) to a model originally proposed by Simo. The goal was to predict the behavior of high-elongation propellants. Swanson's softening function was used with Simo's Cauchy-Green based constitutive law. The influence of particle debonding or vacuole formation was included through a function related to the bulk modulus and the invariants of the strain tensor. Like Burke's model, additional characterization tests were required to calculate the constants found in functions.

Ravichandran and Liu (Ref. 39) recognized the importance that particle debonding had on the behavior of particle reinforced composites. They cited the micromechanical studies of Anderson and Farris (Ref. 3), Mochida (Ref. 40) and Mori and Tanaka (Ref. 28) to support the approach they took to formulate a damage-based constitutive model. Damage was described through an internal scalar variable that was linked to maximum dilatation. The damage function was characterized using volume dilatation data from uniaxial tests.

It can be seen from the review of the above literature that there has been a tendency in the last few years for phenomenological models to include micromechanical processes like particle debonding. It has been recognized, though, that micromechanical and phenomenological techniques are complementary approaches. The micromechanics provide more insight into the physical processes that control behavior while phenomenology provides better computational efficiency (Ref. 35).

Many of the micromechanical models applicable to solid propellants in particular and particulate composites in general have been developed in the past ten years. With Weng's development of a closed-form technique to calculate the effective modulus of an anisotropic composite in 1984 (Ref. 41), an entirely new generation of micromechanical

models have appeared in the literature. The following section summarizes the current research taking place in micromechanical modeling and shows how they lead to the objectives set out for this report.

In 1988, Tandon and Weng (Ref. 42) developed an approximate micromechanical technique to predict the elasto-plastic behavior of particulate composites like silica reinforced epoxy. The analysis was based on Weng's earlier work on the prediction of particulate composite elastic properties using Eshelby's solution of an ellipsoidal inclusion and Mori-Tanaka's (M-T) concept of average stress and strain (Ref. 41). Their analysis examined a particulate composite's response to monotonic proportional loading. This allowed them to use secant moduli to characterize the weakening constraint power of the matrix as opposed to earlier analyses that used an additional eigenstrain term. It also permitted the solution to be cast in terms of deformational theory. Inclusions were assumed to be well dispersed and perfectly bonded in the matrix.

Qiu and Weng (Ref. 24) continued the work from Ref. 42 to include porous composites. They further justified the use of a matrix secant moduli in the M-T theory by pointing out that it allowed one to use the results of many well-developed linear theories to approximate, in a tractable manner, nonlinear behavior. Qiu introduced changes in the way effective stress was calculated to allow prediction of pore expansion under a hydrostatic stress. The effects of spherical and spheroidal voids on composite behavior could be calculated by means of the M-T technique.

Following (Ref. 42), Tzeng developed an M-T based micromechanical model to analyze short fiber/whisker reinforced unidirectional composites, such as SiC whisker reinforced aluminum, undergoing elastic/plastic multiaxial deformation (Ref. 43). He stated

that the secant moduli approach in Ref. 42 allowed a better treatment of strain hardening in the matrix than earlier methods. However, the legitimacy of the M-T technique could be suspect at high volume fractions. The analysis assumed the inclusion to deform elastically while the matrix deformed plastically.

Liming (Ref. 44) used the Eshelby equivalent inclusion method and the M-T technique to analyze nonlinear elastic and viscoelastic particulate composites. Under the assumptions that the matrix Poisson ratio was constant and the inclusions were rigid, he showed that the disturbed strain and the eigenstrain in the Eshelby method could still be related through the Eshelby tensor for a nonlinear matrix. He also showed that it was justifiable to approximate nonlinear behavior through a linear substitution method where the strains in a constitutive equation could be replaced with strains modified by an Eshelby-based tensor. Nonlinear elasticity was modeled through a third-order polynomial based on strain and viscoelasticity was quantified through the Duhamel integral. No experimental data was presented to support the theoretical work.

In 1994, Ju and Chen (Ref. 45) presented a framework to predict the elasto-plastic behavior of a two-phase particle-reinforced metal matrix composite. The inclusions were assumed to remain elastic while the matrix could be elastic or plastic depending on the local stress and deformation. In their analysis, in addition to accounting for a composite's dependency on constituent phase properties, volume fraction and inclusion micro-geometry, they were able to incorporate a technique to account for the effect that inclusion interaction had on overall mechanical behavior. In previous studies (Refs. 29, 46), they showed the M-T solution was valid for composites containing dilute concentrations of inclusions. With an interacting solution, composite properties could be accurately predicted for materials containing up to an inclusion volume fraction of 0.45. Inclusions were assumed to be

uniformly sized and perfectly bonded to the matrix.

Chen et al. (Refs. 47, 48) studied the nonlinear behavior of a particulate reinforced Mooney-Rivlin rubber composite using the M-T method. It was shown how secant moduli appropriate for inclusion in a M-T formulation could be derived from a Mooney-Rivlin description of the matrix. Since they assumed inclusions were perfectly bonded, nonlinearity came strictly from the nonlinear matrix behavior. Predictions were compared to experimental data for glass bead/hydroxl-terminated polybutadiene and glass bead/silicone composites filled to a volume fraction of 0.20.

A recent article by Favier et al. (Ref. 49) addressed the issue of micromechanic damage due to interaction of the matrix with inclusions. They were interested in simulating the nucleation and evolution of damage in two-phase metallic composites caused by inclusion fracture or decohesion. Inclusion fracture was determined by comparing the inclusion stress to a critical fracture stress. Decohesion was detected by comparing the inclusion normal stress to a critical decohesion stress. Once an inclusion was damaged, zero stiffness was assigned to the particle in the direction of stress and lateral strains were set to zero. This effectively created an anisotropic particle. The Eshelby equivalent inclusion method along with an elasto-plastic self-consistent model was used to study damage in a precipitate reinforced Al3004-H19 metal.

Another approach was taken by Zhao and Weng (Ref. 50) to model damage due to inclusion debonding in a two-phase elasto-plastic composite containing aligned oblate inclusions. The study focused on the behavior of a hydrostatically loaded metal matrix composite as a function of inclusion shape, inclusion concentration and interfacial strength. The procedure for calculating overall stress state followed the work of Qiu (Ref. 24). The analysis

assumed a weak interfacial tensile strength so that matrix cracking could be neglected. Debonding was parameterized using a Weibull statistical function so that the probability of debonding was calculated as a function of hydrostatic stress. Interfacial strength was characterized by the Weibull scale and shape parameters. Yamada et al. (Ref. 51) also used a Weibull function to describe debonding but they made their function dependent on particle diameter too. Model performance in (Ref. 50) was evaluated using a hypothetical boron-aluminum metal matrix composite.

The Eshelby and Mori-Tanaka based analyses provide a new avenue to describe the relationships between the components in a particulate composite and the resulting mechanical behavior. As shown, Favier et al. (Ref. 49) used anisotropic properties to model debonded inclusions. Ju and Chen (Ref. 45) improved the Mori-Tanaka predictions by accounting for particle interaction. Tandon and Weng (Ref. 42) showed that matrix secant modulus could be used to characterize the weakening constraint power of the matrix. Liming (Ref. 44) introduced matrix nonlinearity through a nonlinear elastic representation. These concepts form the basis of a new micromechanical model for particulate composites.

3.0 NONLINEAR ELASTIC MICROMECHANICAL MODEL

This section will outline the equations that define the nonlinear elastic micromechanical model. The development will begin with a statement of the governing energy equation and then move to a description of improved M-T modulus prediction routine. The implementation of the nonlinear matrix properties will then be discussed.

3.1 Governing Energy Equation

For any deformable material, the state of stress and strain in a structure can be calculated for a system of loads or displacements using the first law of thermodynamics. In the case of particulate composites, external work is not only stored as internal strain energy but is dissipated through the process of particle debonding. It was shown in Refs. 3 and 25, that this statement could be expressed as

$$2G_c\delta A/V_o = \sigma_{ij}\delta\epsilon_{ij} - \delta\sigma_{ij}\epsilon_{ij} \qquad [1]$$

where G_c is the adhesion energy between particle and matrix, δA is the variation or creation of surface area through debonding, σ_{ij} is the composite stress, ϵ_{ij} is the composite strain and V_o is the specimen volume.

By using the boundary conditions for a uniaxial bar under tension (11-direction) and ambient pressure, the constitutive equation for an orthotropic material can be shown to be (Ref. 26)

$$\sigma_{11} = \left(\overline{C}_{1111} - \frac{2\overline{C}_{1122}\overline{C}_{2211}}{\overline{C}_{2222} + \overline{C}_{2233}}\right)\epsilon_{cr} \qquad [2]$$

where σ_{11} is the true stress in the loading direction and ϵ_{cr} the uniaxial critical strain. Critical strain is defined to be the point where the internal strain energy in the composite and the energy dissipated through particle debonding equals the work put into the composite. In this report, all stress measures are in terms of true stress while strains are defined in terms of engineering strain. This distinction is particularly important when comparisons are made between model results and experimental data in Sec. 5.1. It can be seen from eq. 2 that the average composite tensile modulus E_c is defined by

$$E_c = \overline{C}_{1111} - \frac{2\overline{C}_{1122}\overline{C}_{2211}}{\overline{C}_{2222} + \overline{C}_{2233}} \qquad [3]$$

By differentiating σ_{11} with respect to the bonded particle concentration, c_i, and substituting it and eq. 2 into eq. 1 gives

$$2\frac{G_c}{V_o}\frac{dA}{dc^i} = \left[-\frac{d}{dc^i}\overline{C}_{1111} - 2\frac{\overline{C}_{1122}\overline{C}_{2211}}{(\overline{C}_{2222}+\overline{C}_{2233})^2}\left\{\frac{d}{dc^i}\overline{C}_{2222}+\frac{d}{dc^i}\overline{C}_{2233}\right\}\right. \quad [4]$$
$$\left. +2\frac{\overline{C}_{2222}+\overline{C}_{2233}}{(\overline{C}_{2222}+\overline{C}_{2233})^2}\left\{\overline{C}_{2211}\frac{d}{dc^i}\overline{C}_{1122}+\overline{C}_{1122}\frac{d}{dc^i}\overline{C}_{2211})\right\}\right]\epsilon_{cr}^2$$

This equation assumes that the representative volume element (RVE) is larger than the largest particle so that average stress, strain and moduli can be used. Equation 4 can also be summarized by

$$d\mathcal{E}_{area} = d\mathcal{E}_{mod} \quad [5]$$

where $d\mathcal{E}_{area}$ is the energy dissipated through surface creation and is equal to the lefthand side of eq. 4 and $d\mathcal{E}_{mod}$ is the net internal strain energy due to modulus degradation and is equal to the righthand side of eq. 4.

3.2 Relationship Between Surface Area and Inclusion Concentration

The relationship between increase in surface area due to a decrease in bonded particle concentration can be shown to be (Ref. 52)

$$\frac{dA}{dc^i} = -2\cdot\frac{3(1-\sin\theta)}{r}V_o \quad [6]$$

where r is the particle radius. The factor 2 in eq. 6 accounts for the fact that when a particle debonds, two new surfaces are created. The $\sin\theta$ term has been included to leave open the possibility that a particle may partially debond θ degrees from the equator up to the pole. When $\theta = 0°$, dA/dc^i reaches a maximum of $-6V_o/r$.

The types of particles used for this study were characterized with a log normal size distribution (Ref. 25) defined by

$$\log r = \log\overline{r} + n \quad [7]$$

where r is the particle radius, $\bar{r}$ is the mean particle radius and n is the log standard deviation.

3.3 Elastic Properties of a 3-Phase Composite

Equation 4 requires that a relationship between the volume fraction of the constituent phases and the average composite modulus be known. Initially, a particulate composite can be essentially considered as a material containing only two distinct phases as long as the initial void fraction is negligible. However, at some critical stress level, σ_{cr}, enough energy is input into the material system so that particles begin to debond. According to eqs. 4 and 6, the larger the particle, the lower the energy required for it to debond. When debonding occurs, the composite changes from a two-phase composite containing well-bonded particles and a matrix to a three-phase composite containing well-bonded particles, debonded particles and a matrix. Modeling debonded particles by a vacuole representation gives rise to orthotropic composite properties because the stiffness in the direction of loading is lower than the stiffness perpendicular to the load.

The average elastic properties for a 3-phase composite $[\overline{C}]$ containing well-bonded particles with properties $[C^i]$, matrix with properties $[C^o]$ and vacuoles with properties $[C^v]$ was derived in Ref. 26 to be

$$[\overline{C}] = [C^o] \cdot ([I] \; + \; c^i[\Gamma^i](c^i[I - S^i - \Gamma^i] + [S^i] + [A] \\ + c^v[I - S^v - \Gamma^v] \cdot [S^v + B]^{-1} \cdot [S^i + A])^{-1} \\ + \; c^v[\Gamma^v](c^v[I - S^v - \Gamma^v] + [S^v] + [B] \\ + c^i[I - S^i - \Gamma^i] \cdot [S^i + A]^{-1} \cdot [S^v + B])^{-1}) \qquad [8]$$

where brackets denote square matrices and

$$\begin{array}{lcl} [A] & = & [C^i - C^o]^{-1} \cdot [C^o], \\ [B] & = & [C^v - C^o]^{-1} \cdot [C^o], \\ c^r & = & \text{volume fraction of phase-r}, \\ i & = & \text{parameters relating to inclusions}, \\ v & = & \text{parameters relating to vacuoles}. \end{array}$$

The Eshelby tensor $[S]$ is dependant on the matrix Poisson ratio ν_o and the inclusion shape. $[S]$ is defined by Ref. 29 as

$$S_{ijkl} = \frac{1}{15(1-\nu_o)}((5\nu_o - 1)\delta_{ij}\delta_{kl} + (4 - 5\nu_o)(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})) \qquad [9]$$

Vacuoles have been modeled as a spherical inclusion with orthotropic properties. For a uniaxial bar in tension, a low modulus value, $F_b \cdot E_{11}$, in the loading or pole direction was used to represent the debonded condition and a high or inclusion modulus value E_{22} and E_{33} in the equator direction was used along with setting $\nu_{12} = \nu_{21} = \nu_{13} = \nu_{31} = 0$ to enforce the lateral constraint condition. The factor F_b was introduced to allow for the possibility that a particle could be partially debonded. Since the M-T formulation can be applied equally well to inclusions with orthotropic properties as to inclusions with isotropic properties, this approach was implemented by modifying the definition of the debonded particle's material matrix. The property matrix for the normal components of this orthotropic material was

$$[C^v] = m \begin{bmatrix} F_b \cdot E_{11}(1 - \nu_{23}\nu_{32}) & 0 & 0 \\ 0 & E_{22} & E_{22}\nu_{32} \\ 0 & E_{33}\nu_{23} & E_{33} \end{bmatrix} \qquad [10]$$

where

m = $(1-\nu_{23}\nu_{32})^{-1}$,
F_b = partial debonding factor,
E_{ii} = isotropic tensile modulus of particle in the ii-direction,
ν_{ij} = Poisson's ratio of particle in the ij-direction.

The solution in Ref. 29 for particle interaction was integrated into the M-T formulation through a correction matrix, $[\Gamma^r]$. This matrix was derived from the analysis of probabilistic pairwise particle interaction of two identical and randomly located elastic spheres embedded in a comparison material. It was shown in Ref. 26 to be

$$[\Gamma^r] = [I] + \frac{5c^r}{4\beta^2} Y [W^r] \qquad [11]$$

where

$[I]$ = identity matrix,
c^r = volume fraction of phase-r,
Y = $Y_m(1-c^r)$,
Y_m = interaction factor multiplier,
$[W^r]$ = $\zeta_1 \delta_{ij}\delta_{kl} + \zeta_2(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$,

The definitions for β, ζ_1 and ζ_2 may be found in Ref. 29. The term $[\Gamma^r]$ states that reinforcement not only comes from the presence of inclusions in the composite but that it also occurs due to the proximity of the inclusions to one another. The strength of the interaction is characterized by Y_m.

3.4 Nonlinear Elastic Matrix Behavior

The average composite modulus $[\overline{C}]$ (eq. 8) is controlled by the nature of the constituent properties. If $[C^i]$ and $[C^o]$ are isotropic but $[C^v]$ is orthotropic, the average modulus will necessarily be orthotropic. If the $[C^i]$ and $[C^v]$ are linear elastic but $[C^o]$ is nonlinear elastic, $[\overline{C}]$ will necessarily be nonlinear elastic.

Nonlinear elasticity has been introduced in the current micromechanical formulation by modeling the matrix as an isotropic strain dependent material. This dependency was quantified by the second-order polynomial

$$E_o = a_0 + a_1\epsilon_o + a_2{\epsilon_o}^2 \qquad [12]$$

where E_o is the tensile secant modulus of the matrix and ϵ_o is the average uniaxial matrix strain. The matrix Poisson's ratio, ν_o, was assumed to be constant over the strain range of interest. The E_o and ν_o were converted to strain dependent bulk K_o and shear G_o secant moduli (Ref. 53). These were then used to calculate the elements in the matrix secant modulus $[C^o]$ according to standard relationships (Ref. 54). Since $[C^o]$ is defined in terms of a secant modulus, $[\overline{C}]$ is also a secant measure.

For a particulate composite containing an arbitrary number of phases, the average matrix strain was estimated using a Reuss model (Refs. 55, 56). The resulting relationship between ϵ_o and the average composite strain ϵ_c for a one-dimensional case was (Ref. 57)

$$\epsilon_o = \frac{E_c}{E_o}\epsilon_c = M_\epsilon\epsilon_c \qquad [13]$$

where E_c is the average composite secant modulus and M_ϵ is defined as the composite strain magnification factor. The fact that E_c, E_o and ϵ_c are functions of ϵ_o automatically means an iterative solution is required to solve eq. 8.

3.5 Algorithm for Prediction of Mechanical Behavior

In order to predict the mechanical behavior of a particulate composite using eqs. 2 to 13, five things must be known before the critical strain, ϵ_{cr} and the corresponding average composite stress, σ_c can be calculated. They are:

1. the size distribution of the particles in the composite,

2. the particle and matrix properties,

3. the critical stress where particle debonding first begins,

4. the degradation in composite stiffness as a result of particles becoming debonded,

5. the adhesion energy between particle and matrix.

Particle size distribution and constituent properties can be measured using standard measurement techniques (Ref. 25). The technique to evaluate modulus degradation was discussed in Sec. 3.3. Estimation of adhesion energy will be dealt with in Sec 4.3. Discussions on critical stress and further comments on adhesion energy are given in Sec. 5.1.2.

When the above items have been quantified, the prediction of composite mechanical behavior proceeds in two steps. Up to the critical stress σ_{cr}, before any debonding has occurred, the nonlinear stress-strain relationship for a one-dimensional case maybe calculated according to

$$\epsilon_c{}^{i+1} = (E_c^i)^{-1} \cdot \sigma_c^j \qquad [14]$$

using the method of direct substitution (Ref. 58). Here, i is the iteration index, σ_c^j is the j-th composite stress ($1 < j < M$, at $j = M$, $\sigma_c^M = \sigma_{cr}$), E_c^i is the average composite properties evaluated at ϵ_c^i and ϵ_c^{i+1} is the new estimate of ϵ_c.

After the critical stress has been reached, the solution continues with the debonding of the k-th group of particles out of a total of K groups starting from the largest particle radius. Particle sizes in each group are calculated according to eq. 7. This determines the energy dissipated through new surface area creation ($d\mathcal{E}_{area}$ in eq. 5) and the concentration of inclusions that have debonded. Calculation of the net change in average composite

modulus (bracketed quantity on the righthand side of eq. 4) is carried out using the modulus degradation routine from Sec. 3.3. From this, the critical strain is determined. The corresponding stress is then calculated using eq. 2 and the composite properties $[\overline{C}]$ which lead to the critical strain. Matrix strain is updated along with each critical strain calculated (eq. 13). As the process of debonding groups of inclusions carries on, the average mechanical behavior of the composite is described by the pairs of critical strain-composite stress points.

4.0 EXPERIMENTAL

4.1 Materials

A model composite material was fabricated using hydroxyl-terminated polybutadiene (HTPB) and spherical glass beads. The HTPB (Arco Chemicals) had a nominal molecular weight of 2800 g/mol, a polydispersity index of 1.8 and a hydroxyl equivalent weight of 40.2 mg KOH/g. Four types of glass beads were purchased from Potter's Industries Inc. (La Prairie, Qc, Canada) for the experiments. The first type was as-received beads with an advertised average diameter of 25 μm (Stock 2900). The second type was as-received beads with an advertised average diameter of 100 μm (Stock 2227). The third and fourth types were Stock 2900 and Stock 2227 treated with a silane coupling agent known as CP-03. This agent is optimized for use with epoxy and urethane resin systems.

The test matrix given in Table I was designed to examine the performance of the micromechanical model given different particle size distributions, adhesion energies, inclusion volume fractions and loading rates. Each composite designation is composed of four letters. The first letter indicates the type of surface treatment (N-untreated, T-treated).

The second letter indicates the initial inclusion volume fraction (3-30%, 5-50%). The third letter identifies the crosshead displacement rate used in the tensile test (M-10 mm/min, F-100 mm/min) and the fourth letter indicates the average bead size (S-25 μm, L-100 μm).

The model composite was fabricated in two steps. A pre-mix was prepared for casting by mixing in 0.5% w/w AO2246 (Cyanamid) anti-oxidant agent using a Design Integrated Technologies 10CV helicone vertical mixer. Mixing time was 2 hours at 60°C. The polymer was then left to stand under vacuum for 2 days at 60°C to ensure that trace moisture had been eliminated. On the day of casting, 0.01% w/w di-ter-butyl-dilaurate (DBTDL) cure catalyst from Aldrich Chemical was added to the pre-mix along with the required quantity of glass beads and mixed for 30 minutes. To achieve an optimal NCO/OH ratio of 1.1, 6.55% w/w tolylene diisocyanate (TDI, comprised of 97% 2,4 and 3% 2,6 isomers) from Kodak Ltd. was added and the entire mixture was mixed again for 30 minutes. At the end of the mix cycle, the composite was cast into 150 mm $\times$ 150 mm $\times$ 100 mm blocks for specimen preparation. All mixing and casting operations took place under vacuum. The blocks were left to cure for 6 days at 60°C under ambient pressure.

4.2 Test Procedures

Uniaxial specimens were prepared by sawing the composite blocks into 12.5 mm slabs and then die cutting JANNAF Class C type specimens from the slabs. The 9.5 mm $\times$ 12.5 mm cross-sectional area made this specimen well-suited for mechanical characterization of loaded materials (Refs. 25, 59). Prior to testing, the specimens were pre-conditioned in a vacuum desiccator at room temperature for at least 24 h.

Uniaxial testing was carried out on the composites in an INSTRON 4206 machine

equipped with either an OPTRA Laser Extensometer (OPTRA Inc., Peabody, MA) or a Farris Gas Dilatometer (Richard Farris, Leeds, MA) according to CPIA procedures (Ref. 59). One series of tests were carried out with the laser extensometer to calculate the effective gage length (EGL) of the model composites. Details of this procedure were given in Ref. 25. Another series of tests were carried out in the gas dilatometer to simultaneously measure the stress-strain behavior as well as the dilatation-strain behavior. Details of the how this instrument operates is given in Refs. 57 and 60. The EGL was needed for the dilatometer tests because composite strain could only be calculated using crosshead displacement. Three specimens were tested for each combination shown in Table I. Since data scatter was low, the test best representing the average behavior of the three specimens was selected for use in Sec. 5.0. Tensile tests on pure polymer were conducted according to ASTM D638 (Ref. 61).

Initial inclusion volume fraction c_o^i was measured using density measurements of pure polymer blocks and composite blocks. Density measurements were carried out using the immersion method as specified in ASTM D792-86 (Ref. 62). Initial volume fraction was calculated using a rearranged form of the rule of mixtures equation for composite density (Ref. 52).

TABLE I

Test matrix for glass bead/polybutadiene composites

Xhead Rate (mm/min)	Treat-ment	avg. c_o^i (%)	Avg. dia. (μm)	Designation
10	none	30	25	N3MS
		30	100	N3ML
		50	25	N5MS
		50	100	N5ML
10	silane	30	25	T3MS
		30	100	T3ML
		50	25	T5MS
		50	100	T5ML
100	none	30	25	N3FS
		30	100	N3FL
		50	25	N5FS
		50	100	N5FL
100	silane	30	25	T3FS
		30	100	T3FL
		50	25	T5FS
		50	100	T5FL

c_o^i is the initial inclusion volume fraction

4.3 Estimation of Adhesion Energy

Adhesive strength has been measured using a number of techniques (Refs. 63 - 65). As noted by Mower (Ref. 66), these techniques evaluated the adhesive strength qualitatively through assumed adhesive characteristics or fractographic evidence. In Ref. 66, the authors evaluated adhesive strength between particle and matrix in terms of a hydrostatic stress using a uniaxial test. Their test specimen consisted of a particle embedded in the center of a matrix bar that was specially shaped to induce a triaxial state of stress at that point.

The approach used here was similar in concept to that used by Mower. However, instead of measuring adhesive strength, adhesion energy was desired. The basis of this measurement came from a simplified form of eq. 4 (Ref. 26),

$$\frac{G_c}{V_o}\frac{\Delta A}{\Delta c_i} = -\frac{1}{2}\cdot\frac{\Delta E_c}{\Delta c_i}\epsilon_{cr}^2 \qquad [15]$$

Referring to Figs. 1 and 2, eq. 15 states that if we load the specimen shown in Fig. 1 in tension, at some point ϵ_{cr}, there would be enough work input into the specimen to cause the glass bead to debond (point A in Fig. 2). When this occurs, there will be a sudden loss in reinforcement and therefore overall stiffness, so the load will fall to point B. If we unload from point B back to zero load (point O), the area OABO represents the energy dissipated to create new surface area. This energy is quantified by the righthand term of eq. 15. Since the geometry of the specimen and inclusion are known and the amount surface area debonded is observed during the test, adhesion energy G_c can be calculated.

The shape of the specimen shown in Fig. 1 was designed using a finite element model to concentrate the stress field around the pole of the glass bead and to minimize the peel stresses at the edges of specimen grips. The specimen volume was selected according to the size of beads available so that the $\Delta E_c/\Delta c_i$ would be measurable in terms of a drop

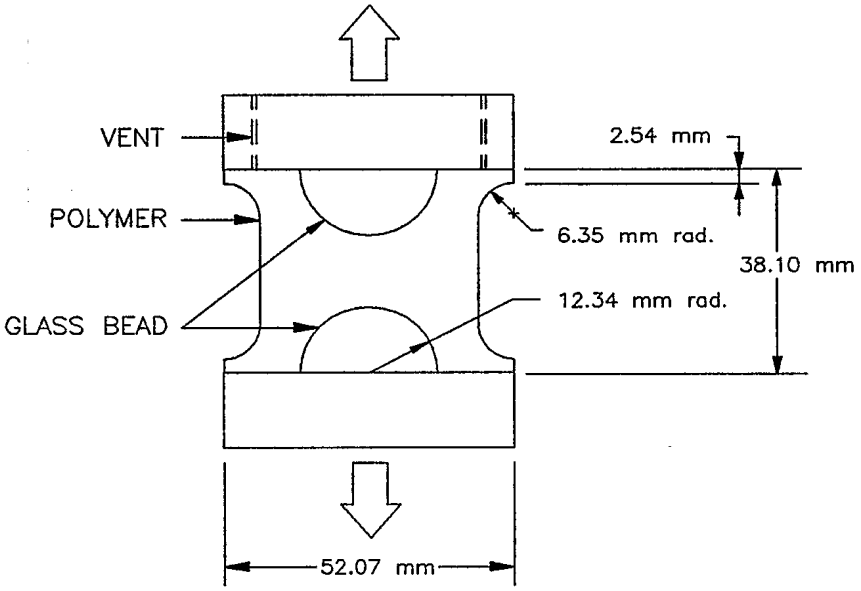

FIGURE 1 – Geometry of test specimen used to measure adhesion energy

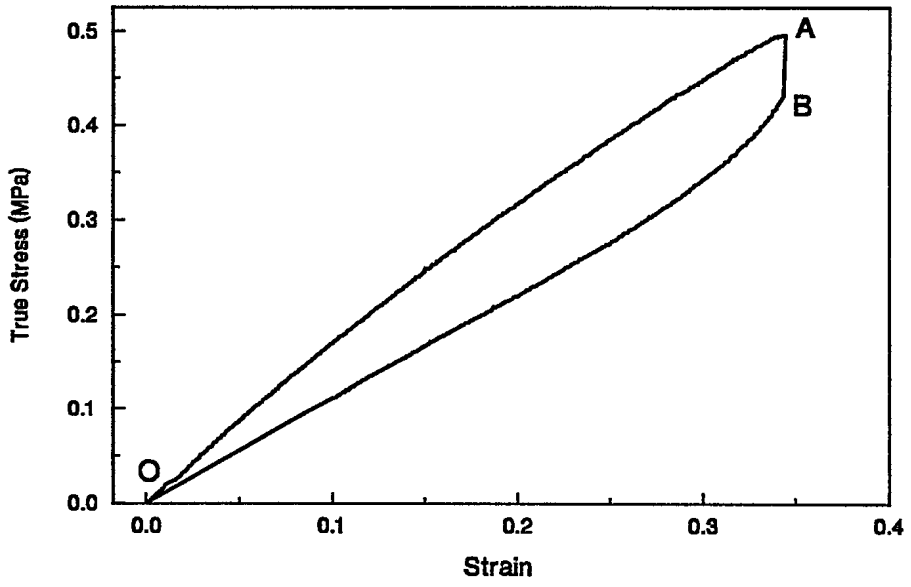

FIGURE 2 – Schematic of energy loss due to particle debonding

in load at a reasonable crosshead displacement. The input file for the ANSYS finite element program may be found in Appendix A.

Six specimens were fabricated using ordinary soda-lime beads (ie. glass marbles) and HTPB polymer with a NCO/OH ratio of 1.0. The NCO/OH ratio was lowered slightly to better match the matrix modulus measured using specimens from the cast blocks in Sec. 4.1. The polymer modulus was verified using specimens that were fabricated without beads.

To fabricate the specimens, the beads and grip surfaces were degreased using dichloromethane. The lower grip was then set upright and the clamshell Teflon molds conforming to the geometry shown in Fig. 1 clamped on. The degassed polymer was poured into the mold while it was still at 60°C. Afterwards, the top grip was pressed on and the excess polymer allowed to exit through the vents. This created essentially void-free specimens. Any air bubbles that did become trapped were usually small and located in the low stress areas in the specimen. These had negligible effect on the overall results.

The specimens were tested at two loading rates. Three specimens with beads and one specimen without beads were tested at a crosshead rate of 10 mm/min. The remaining specimens were tested at a rate of 100 mm/min. The purpose of these experiments was to obtain an idea of the relative change in adhesion energy for this type of composite at the two test speeds. No attempts were made to characterize the actual adhesion energy that existed between as-received glass beads or CP03-treated beads and HTPB. Therefore, the values estimated with these single-bead tests are not meant to be a representative measure of the adhesion energy present in the composites fabricated in Sec. 4.1 since the surface treatments are obviously different.

5.0 ANALYSIS AND DISCUSSION

The micromechanical model described in Sec. 3.0 contains adjustable parameters that must be defined before the model can be run. For example, the user must specify the appropriate value for the partial debonding factor, F_b (eq. 10). Questions of how one selects these values and how sensitive the predictions are to small changes in these values naturally arise.

The first question will be dealt with in Sec. 5.1. Here the experimental data will be used to deduce the values for the adjustable parameters. This analysis will also allow critical examination of the assumptions and theory presented in Sec. 3.0. The second question will then be examined in Sec. 5.2 in light of the parameters obtained in Sec. 5.1. In this section, since uniaxial tests were carried out, the notation used will refer to scalar values of tensile secant modulus E, stress σ and strain ϵ. Also, superscript "e" will be used to denote experimental data while superscript "c" will be used to denote calculated results. Comparisons will be made at discrete points so the differential operator has been replaced by a Δ operator to reflect this.

5.1 Determination of Model Parameters

5.1.1 Procedure

The known quantities in the model are the experimentally measured composite engineering stress σ_c^{eng}, composite strain ϵ_c^e, dilatation $\Delta V^e/V$, initial inclusion fraction c_o^i, particle distribution ($\overline{r}$ and n in eq. 7), isotropic matrix properties E_o and ν_o, and isotropic particle properties, E_i and ν_i. The experimental true stress σ_c^T and secant modulus E_c^e can

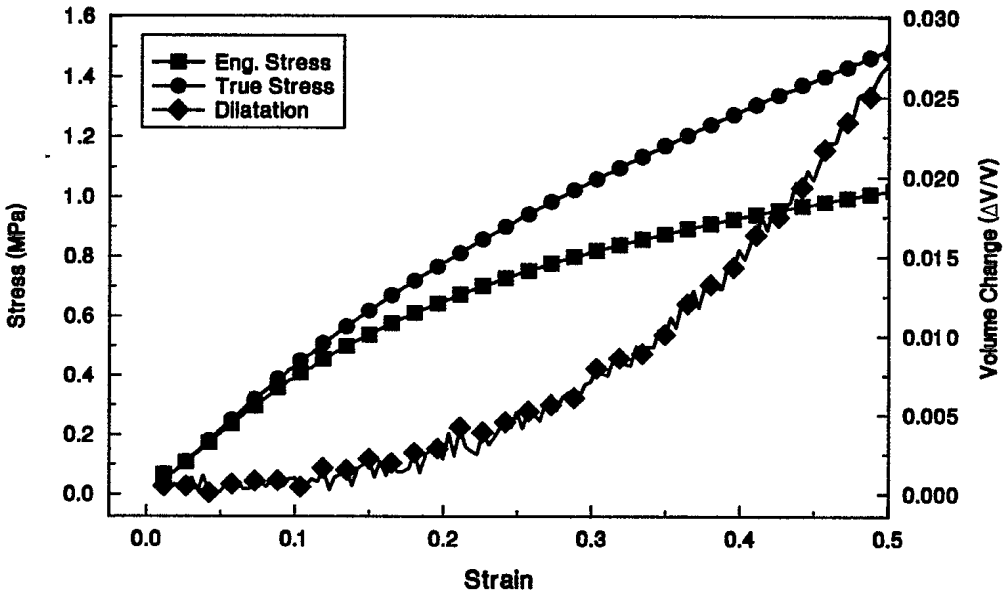

FIGURE 3 – Experimental mechanical behavior of composite T3FS

be calculated through the relationships (Ref. 57)

$$\sigma_c^T = \sigma_c^{eng} \cdot \frac{1+\epsilon_c^e}{1+\Delta V^e/V} \quad [16]$$

$$E_c^e = \sigma_c^T \cdot (\epsilon_c^e)^{-1} \quad [17]$$

An example of the experimental data is shown in Fig. 3 for composite T3FS. As the composite is strained, no significant change in volume occurs until $\epsilon_c \approx 0.12$. The stress corresponding to this strain was defined in Sections 3.3 and 3.5 as the critical stress σ_{cr}. As strain increased, the cross-sectional area reduced according to the instantaneous composite Poisson ratio. This resulted in the growing difference seen between composite true stress and engineering stress.

The unknown parameters in the model are the interaction factor multiplier Y_m (eq. 11) and the partial debonding factor F_b (eq. 10). Quantities such as the vacuole

volume fraction c_v (eq. 8) and the adhesion energy G_c (eq. 4), even though not measured directly, can be deduced using the experimental data and the micromechanical model. Y_m can be determined using ϵ_c^e and E_c^e data up to σ_{cr}. After debonding occurs, ϵ_c^e, E_c^e and $\Delta V^e/V$ are needed to determine F_b and to deduce c_v.

The algorithm used to determine Y_m is shown in Fig. 4. The routine starts off by assuming $Y_m = 1$. It proceeds by estimating the i-th iteration of matrix strain ϵ_o^i using the j-th pair of $E_c^e - \epsilon_c^e$ points in the data set. This value is used in the M-T routine (eq. 8) to calculate the E_c^c (eq. 3) that corresponds to ϵ_o^i. The same is done for E_o (eq. 12). A new value of ϵ_o^{i+1} is calculated using E_c^c, E_o and ϵ_c^e (eq. 13). If the value of $\epsilon_o^{i+1} \approx \epsilon_o^i$ then the solution for the matrix strain has converged and the j-th pair of calculated points $E_c^c - \epsilon_c^e$ is stored. This continues for the M data points leading up to σ_{cr}. At $j = M$, the relative error between the experimental modulus E_c^e and calculated modulus E_c^c at each ϵ_c^e is calculated. If the average relative error Y_m^{err} between E_c^e and E_c^c is greater than the tolerance, Y_m is adjusted and the entire procedure repeated. When $Y_m^{err} < tol$ the appropriate interaction multiplier for the composite has been found.

The algorithm used to determine F_b is similar to that used for finding Y_m except this time, experimental values for modulus and dilatation are used (Fig. 5). The Y_m previously calculated is assumed to remain constant for the entire loading history. The routine starts off by assuming $F_b = 0$, ie. the inclusion debonds completely with no residual stiffness in the loading direction. A quantity of vacuoles c_v^j are also assumed created as a result of debonding at the j-th pair of $E_c^e - \epsilon_c^e$ and $\Delta V^e/V - \epsilon_c^e$ data points. As before, the solution for ϵ_o^i is allowed to converge. If $E_c^c \neq E_c^e$ then c_v^j is increased until this condition is met.

The corresponding dilatation $\Delta V^c/V$ is calculated by

$$\frac{\Delta V^c}{V} = \left(1 - 2\frac{\overline{C}_{2211}}{\overline{C}_{2222} + \overline{C}_{2233}}\right)\epsilon_c^e \qquad [18]$$

The calculated pairs of $\Delta V^c/V - \epsilon_c^e$ are stored for $j = M + 1$ to N where N is the total number of data points. At $j = N$, the average absolute error F_b^{err} between $\Delta V^c/V$ and $\Delta V^e/V$ is calculated. If the error is greater than the tolerance then F_b is adjusted and the whole procedure repeated until the tolerance value is reached. When $F_b^{err} < tol$, the appropriate debonding factor for the composite has been found. Since a constant F_b is calculated, it represents the average degree of debonding for particles of all sizes in the composite. The FORTRAN program used for the back-calculation of Y_m and F_b may be found in Appendix B.

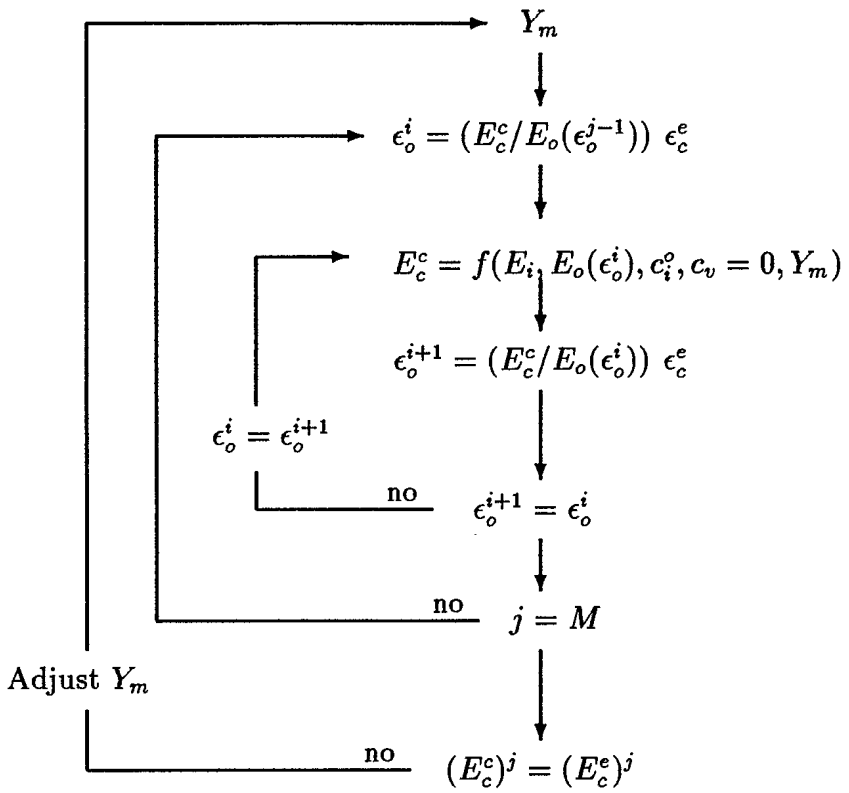

FIGURE 4 – Algorithm for back-calculation of Y_m.

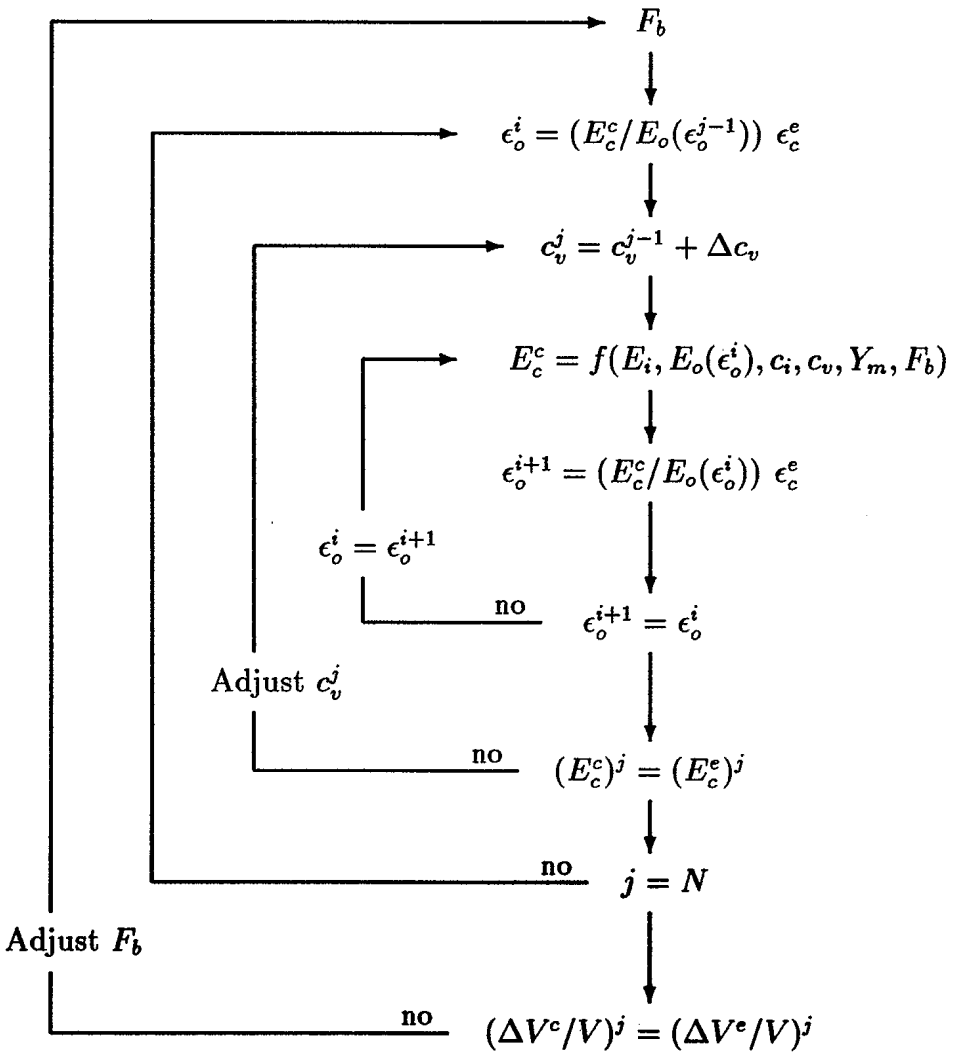

FIGURE 5 – Algorithm for back-calculation of F_b

An apparent adhesion energy G_c^{app} can be calculated from the experimental $E_c^e - \epsilon_c^e$ data and the parameterized particle distribution (eq. 7) by assuming that at the end of data, the total surface area debonded according to eq. 6 and Sec. 3.5 equals the total surface area debonded experimentally (FORTRAN listing in Appendix B). In other words,

$$\sum_{k=1}^{K} \Delta\mathcal{E}_{area}^c = \sum_{j=1}^{N} \Delta\mathcal{E}_{mod}^e \qquad [19]$$

Rearrangement of eq. 19 in terms of eq. 15 gives

$$G_c^{app} = \left(\frac{1}{2}\sum_{j=1}^{N} \frac{\Delta E_c^e}{\Delta c_i}\epsilon_c^{e2}\right) \cdot \left(\frac{1}{V_o}\sum_{k=1}^{K} \frac{\Delta A}{\Delta c_i}\right)^{-1} \qquad [20]$$

The summation of energy for $\Delta\mathcal{E}_{mod}^e$ is straight forward since it is simply the sum of all the energies dissipated at the N experimental data points. The summation for $\Delta\mathcal{E}_{area}^c$ is not as obvious because the total energy depends on the value selected for the total number of points K used to sub-divide the particle distribution. If K is large, the resulting sum of $\Delta A/\Delta c_i$ will be large because there will be many $1/r$ terms to add up. The opposite is true when K is small. The appropriate K was selected by matching the average Δc_i from the particle distribution with the average Δc_i deduced from the experimental data.

5.1.2 Results

The numerical results of the analyses described in Sec. 5.1.1 are shown in the lower part of Tables II and III. The upper part of the Tables show the measured model parameters. From the note included in the Tables, it can be seen that the matrix modulus was slightly nonlinear out to about 100%ϵ for the two crosshead rates used. The tolerances specified for Y_m^{err} ranged from 0.5% to 5% while the tolerances for F_b^{err} ranged from 0.001 to 0.014. Different values were needed because the degree of fit between the experimental and calculated results were not always the same.

TABLE II

Model parameters for glass bead/HTPB tested at 100 mm/min

Composite	T3FS	T3FL	T5FS	T5FL	N3FS	N3FL	N5FS	N5FL
$\overline{r}$ (μm)	15.5	65	15.5	65	15.5	65	15.5	65
n	0.167	0.0374	0.167	0.0374	0.167	0.0374	0.167	0.0374
c_o^i	0.307	0.311	0.511	0.517	0.302	0.302	0.503	0.504
c_o^v	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ν_i	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
ν_o	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
G_i (GPa)	30	30	30	30	30	30	30	30
E_o (MPa)	see a	see a	see a	see a	see a	see a	see a	see a
σ_{cr} (MPa)	0.5	0.5	0.9	0.95	0.35	0.30	0.65	0.8
Y_m	1.5	1.69	0.98	1.18	1.67	1.58	1.14	1.31
$F_b \times 10^{-4}$	2.6	2.1	1.0	0.8	0.8	0.34	1.1	0.42
G_c^{app} (J/m^2)	2.33	5.72	3.98	10.9	2.24	3.83	1.78	5.19

a $E_o = 1.522255 - 0.460286\epsilon + 0.270235\epsilon^2$ (MPa)

$\overline{r}$, average particle radius, n log standard deviation in radius, c_o^i, initial inclusion volume fraction, c_o^v, initial vacuole volume fraction, ν_i, inclusion Poisson ratio, ν_o, matrix Poisson ratio, G_i, inclusion shear modulus, E_o matrix tensile modulus, σ_{cr}, critical stress, Y_m, interaction factor multiplier, F_b, partial debonding factor, G_c^{app}, apparent adhesion energy.

TABLE III

Model parameters for glass bead/HTPB tested at 10 mm/min

Composite	T3MS	T3ML	T5MS	T5ML	N3MS	N3ML	N5MS	N5ML
$\overline{r}$ (μm)	15.5	65	15.5	65	15.5	65	15.5	65
n	0.167	0.0374	0.167	0.0374	0.167	0.0374	0.167	0.0374
c_o^i	0.307	0.311	0.511	0.517	0.302	0.302	0.503	0.504
c_o^v	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ν_i	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
ν_o	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
G_i (GPa)	30	30	30	30	30	30	30	30
E_o (MPa)	see a	see a	see a	see a	see a	see a	see a	see a
σ_{cr} (MPa)	0.45	0.50	0.65	0.75	0.45	0.30	0.40	0.45
Y_m	1.17	1.26	0.83	1.13	1.22	0.92	1.06	1.32
$F_b \times 10^{-4}$	2.1	1.8	1.0	1.1	0.4	0.24	1.0	0.3
G_c^{app} (J/m^2)	1.20	3.65	2.23	6.34	1.43	2.22	1.13	2.91

a $E_o = 1.554865 - 0.497499\epsilon + 0.321452\epsilon^2$ (MPa)

$\overline{r}$, average particle radius, n log standard deviation in radius, c_o^i, initial inclusion volume fraction, c_o^v, initial vacuole volume fraction, ν_i, inclusion Poisson ratio, ν_o, matrix Poisson ratio, G_i, inclusion shear modulus, E_o matrix tensile modulus, σ_{cr}, critical stress, Y_m, interaction factor multiplier, F_b, partial debonding factor, G_c^{app}, apparent adhesion energy.

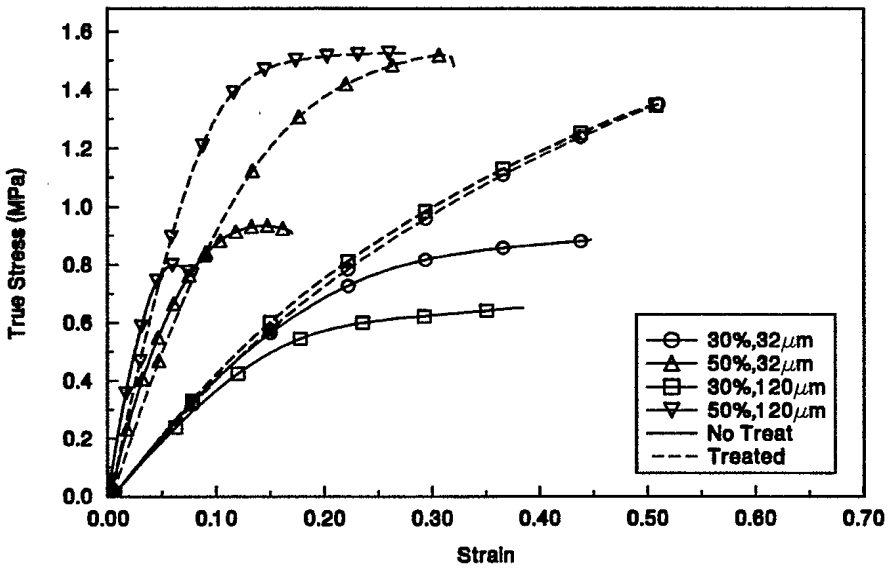

FIGURE 6 – Experimental true stress results for treated (Tr) and untreated (NoTr) glass bead/HTPB composites tested at 10 mm/min

Figures 6 to 9 (p. 33) summarize the tensile and dilatational behavior observed for the composites identified in the test matrix (Table I). Generally, the composites containing the CP-03 treated beads had higher maximum strength and lower dilatation than the equivalent composite which contained as-received beads. The treated beads also delayed the onset of dilatation. For the $c_o^i = 0.3$ composites, the initial modulus was unaffected by surface treatments or bead size. However, for the $c_o^i = 0.5$ composites, slight increases in initial moduli were seen for those composites containing as-received beads.

Figures 10 to 13 (p. 37) show the ability of the micromechanical model to reproduce the experimental data for composites T3FS–N3FS, T5FS–N5FS, T3ML–N3ML and T5MS-N5MS when the parameters from Tables II and III are used. Comparisons for the complete set of composites tested may be found in Appendix C. For most composites, the calculated composite stress σ_c^c matches the experimental stress σ_c^e well up to the end of the data. It

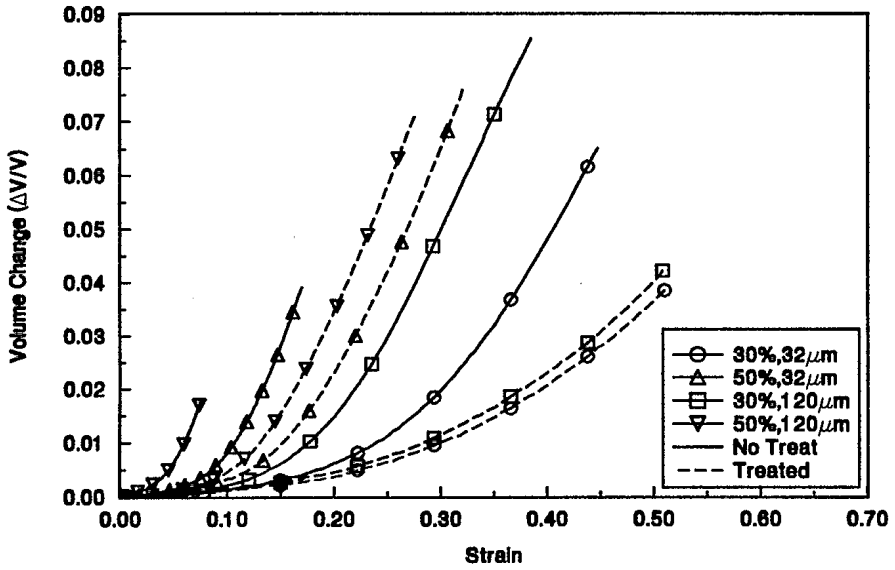

FIGURE 7 – Experimental dilatation results for treated (Tr) and untreated (NoTr) glass bead/HTPB composites tested at 10 mm/min

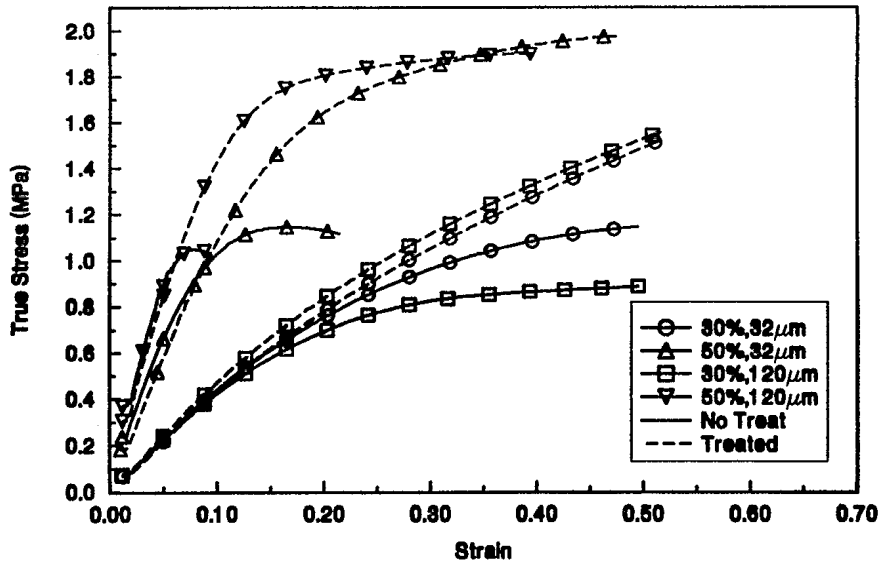

FIGURE 8 – Experimental true stress results for treated (Tr) and untreated (NoTr) glass bead/HTPB composites tested at 100 mm/min

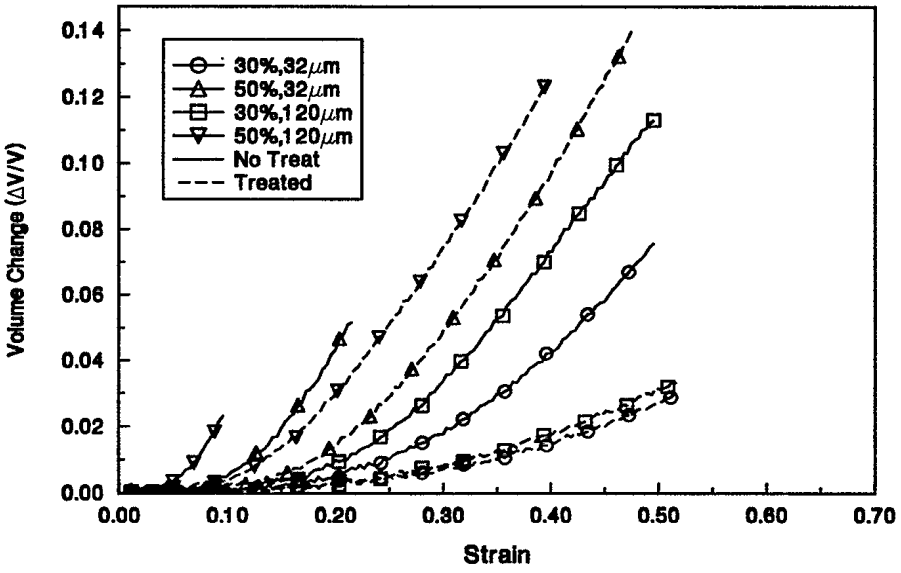

FIGURE 9 – Experimental dilatation results for treated (Tr) and untreated (NoTr) glass bead/HTPB composites tested at 100 mm/min

can be seen that the σ_c^c results have more of a "knee" when compared to the transition seen in the experimental data. The good fit between calculated results and experimental data indicates that the assumption of a constant Y_m was a reasonable one to make.

The deviations in calculated stress were closely related to the deviations of the calculated vacuole fraction c_v^c from the experimental vacuole fraction c_v^e (Fig. 14, p. 39). Since composite stresses are calculated from a current composite modulus that itself is dependent on the current vacuole fraction, it would be expected that the larger the deviation between calculated and experimental c_v, the larger the deviation between the σ_c's. As foreseen, the σ_c^c to σ_c^e difference is opposite in sign and proportional to the c_v^c to c_v^e difference. The "knee" in the $\sigma_c^c - \epsilon_c^c$ curve is related to the apparent lack of strain energy available to debond the first set of particles. For example, in N5MS, debonding starts at $\epsilon_c^e = 0.04$ in the experimental data as opposed to $\epsilon_c^c = 0.06$ in the calculated results. The experimental results

suggest that it may require less internal strain energy to initiate debonding than assumed in the micromechanical model. It should also be mentioned here that the requirement to specify σ_{cr} could be removed if a suitable initiation criterion could be found.

The calculated dilatational behavior ($\Delta V^c/V$) tended to be on the stiff side when compared to the experimental dilatation $\Delta V^e/V$. For cases such as T3FS and T5MS (Figs. 11 and 13), the $\Delta V^c/V$ results compared well with the $\Delta V^e/V$ results. For other cases such as T5FS, N5FS and N5MS, $\Delta V^c/V$ was over estimated at the lower ϵ_c^e and under estimated at the higher ϵ_c^e. For N3ML, the $\Delta V^c/V$ was generally over estimated for the entire strain range while for N3FS, it was under estimated.

The $\Delta V^c/V$ results are controlled both by c_v and F_b. By assuming a value for F_b before determining the c_v that reproduces the E_c^e, the procedure described in Sec. 5.1.1 is only capable of finding an average debonding factor. In cases such as T3FS and T5MS, the assumption of a constant debonding factor appears to work well. In other cases such as T5FS, N5FS or N5MS, it appears that the calculation of F_b needs to be refined in order to capture the debonding behavior of the particles. Unfortunately, with only composite stress, strain and dilatation data on hand, there is not enough experimental information available to resolve this issue.

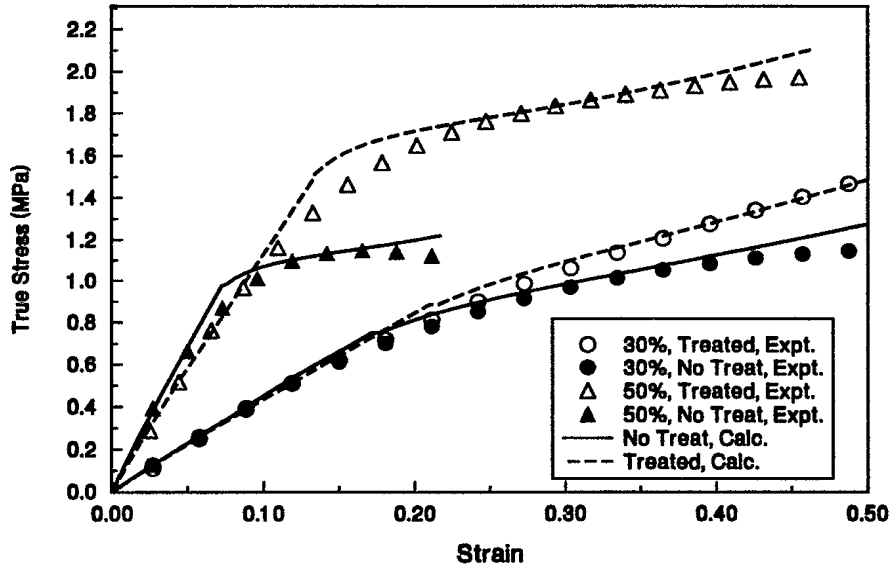

FIGURE 10 – True stress behavior calculations for selected treated (Tr) and untreated (NoTr) glass bead/HTPB composites tested at 100 mm/min

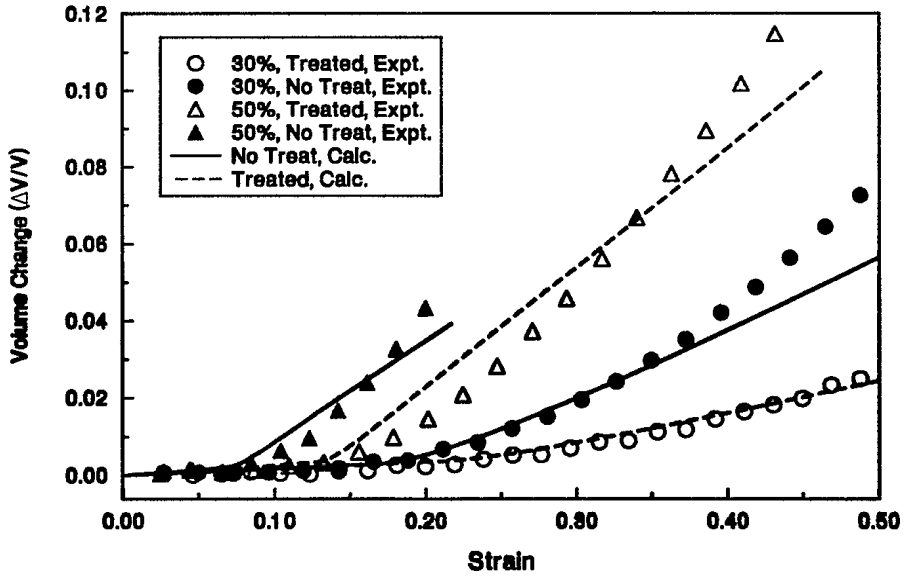

FIGURE 11 – Dilatation behavior calculations for selected treated (Tr) and untreated (NoTr) glass bead/HTPB composites tested at 100 mm/min

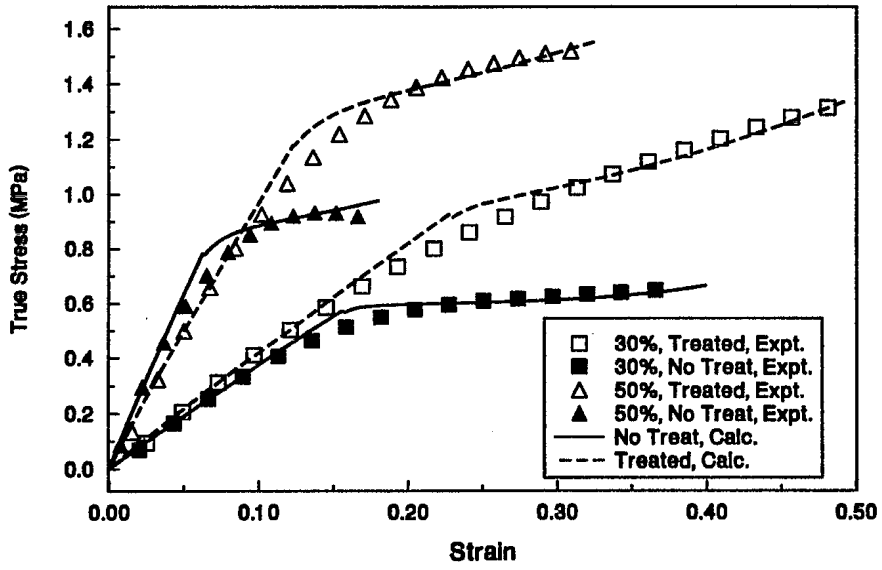

FIGURE 12 – True stress behavior calculations for selected treated (Tr) and untreated (NoTr) glass bead/HTPB composites tested at 10 mm/min

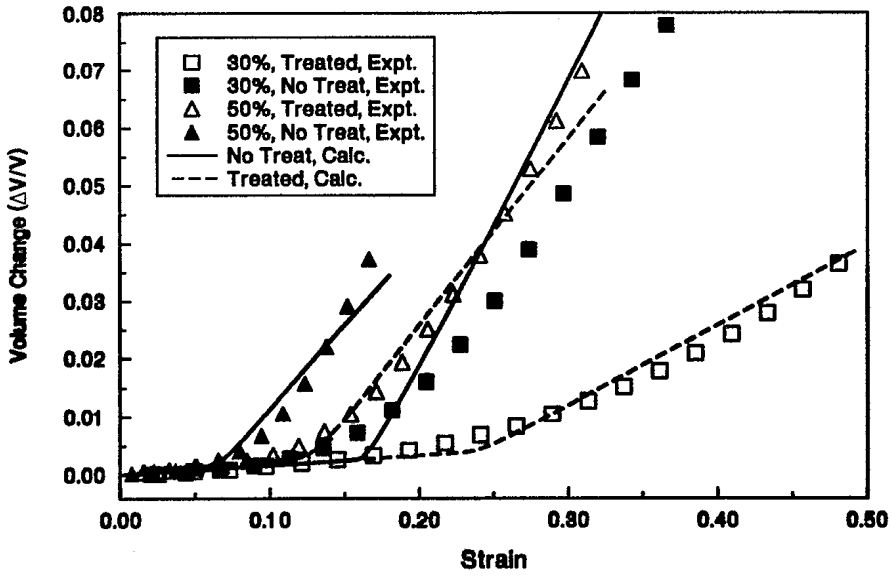

FIGURE 13 – Dilatation behavior calculations for selected treated (Tr) and untreated (NoTr) glass bead/HTPB composites tested at 10 mm/min

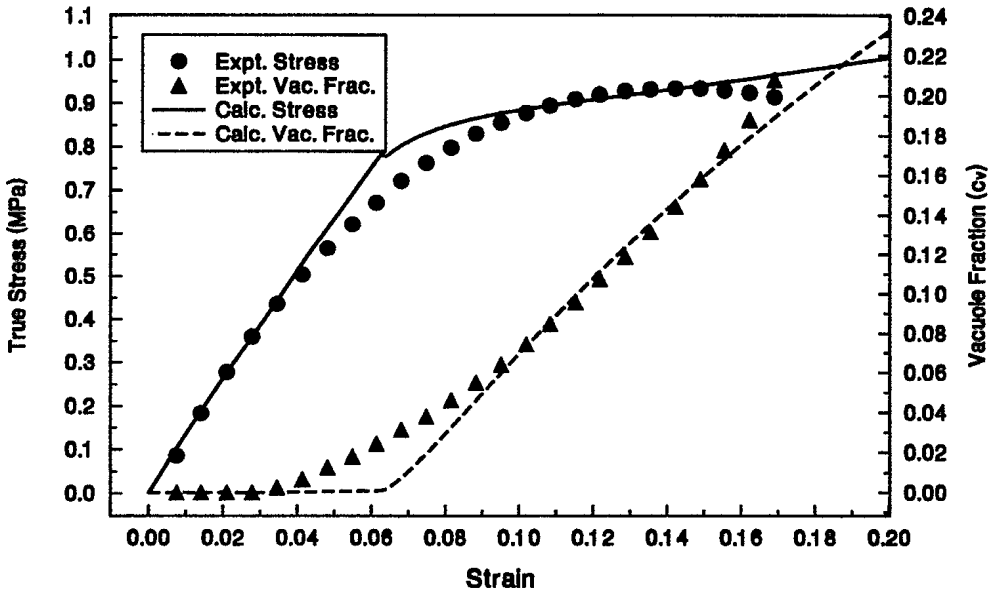

FIGURE 14 – True stress and vacuole concentration calculations for composite N5MS

From Tables II and III, the result $Y_m \neq 1$ for all composites tested indicates that the composite modulus in eq. 8 requires some adjustment to reproduce the actual initial modulus. While it would be tempting to say that the Y_m values are purely attributable to physical particle interaction (Ref. 67), the fact that the Y_m for composites containing $c_o^i = 0.5$ is lower than the Y_m for composites containing $c_o^i = 0.3$, except in the case of N3ML and N5ML, discounts this interpretation. From these results, Y_m can be considered a parameter that groups together factors such as particle interaction, size and surface treatment that have a subtle influence on composite modulus.

Again from the Tables, it can be seen that the values of $F_b \neq 0$. This implies that the beads do not fully debond but that there is a residual bond that remains between bead and matrix. Physically, this would imply that $\theta \neq 0°$ (eq. 6) as assumed by others (Refs. 3, 40, 68, 69) but it would take on some value $\theta > 0°$. This bond is significant since it varies

between 1 to 12 times the stiffness of the matrix. The higher F_b for composites containing 31 μm beads suggests smaller beads debond to a lesser extent than larger beads.

One of the major assumptions made in the micromechanical model dealt with how particles debond as the composite is loaded. Based on the information found in the literature (Refs. 5 - 8), it is generally accepted that large particles debond before smaller ones. This model takes that fact one step further by assuming that the particles in the composite debond in a progressive manner starting with the largest particles and ending with the smallest ones. The possibility that some large particles and some small particles debond at the same time is not considered. The experimental data and calculated results for rate of particle debonding (denoted Δc_v) versus the cumulative c_v for T5MS (Fig. 15, p. 41) and N5MS (Fig. 16), show that the assumption was a reasonable one. This assumption worked well for 14 out of the 16 composites tested. In the cases of T5FL and T5ML, large differences were observed between the deduced Δc_v^e and the calculated Δc_v^c (see Fig. 17 for T5ML results). This explained why there was poor correspondence between the experimental and calculated $\sigma_c - \epsilon_c$ results (Fig. 18). Without other information, it is not possible to determine why the beads in these composites tended to debond at a slower rate than expected.

The apparent adhesion energies listed in Tables II and III indicate that the energy required to debond a particle is rate sensitive and must be taken into account. These values were calculated assuming the debonding angle $\theta = 0°$. On average, when equivalent composites are compared, the G_c^{app} at 100 mm/min was about 1.7 times the G_c^{app} at 10 mm/min.

The results from the adhesion tests (Sec. 4.3) shown in Table IV (p. 43) also support

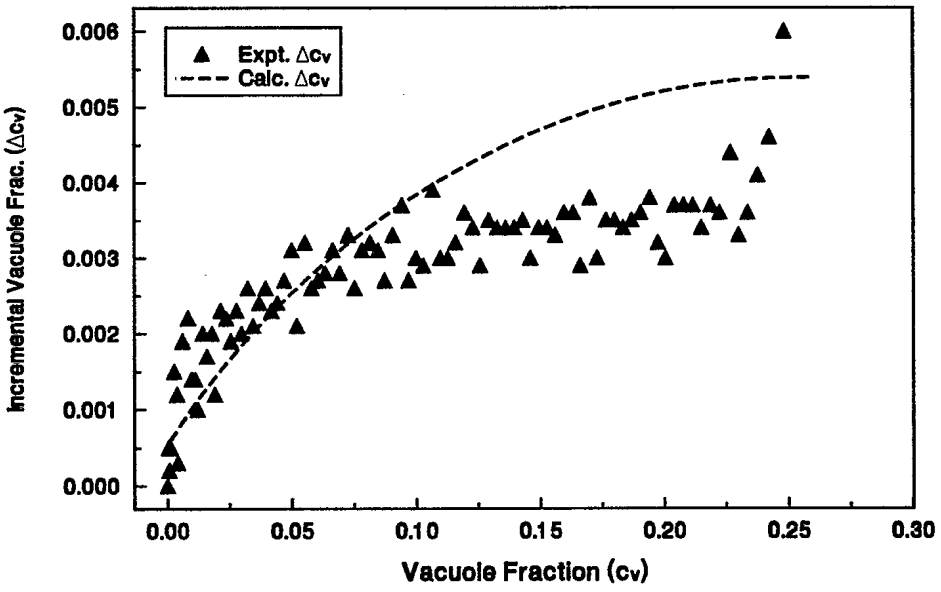

FIGURE 15 – Calculated incremental vacuole concentration behavior for composite T5MS

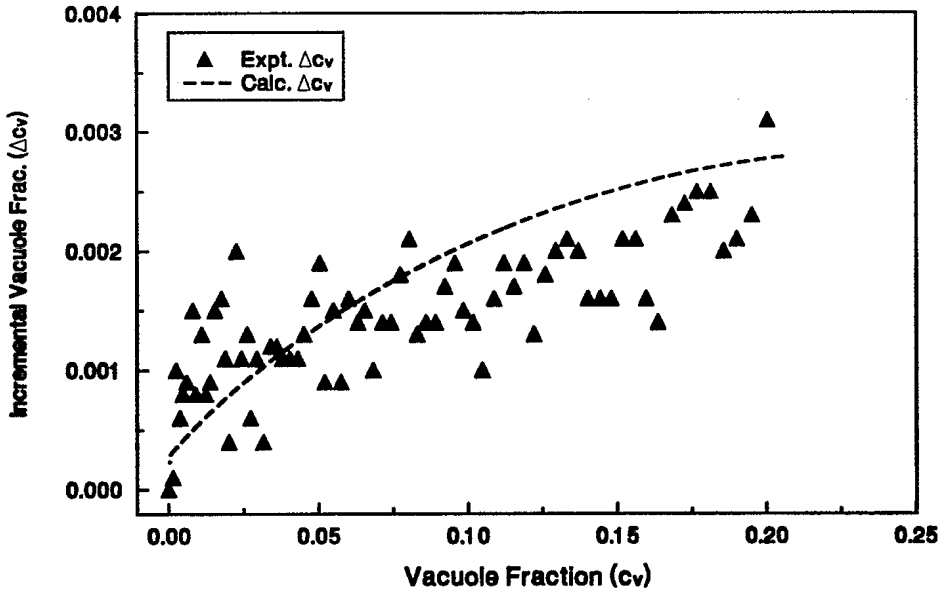

FIGURE 16 – Calculated incremental vacuole fraction behavior for composite N5MS

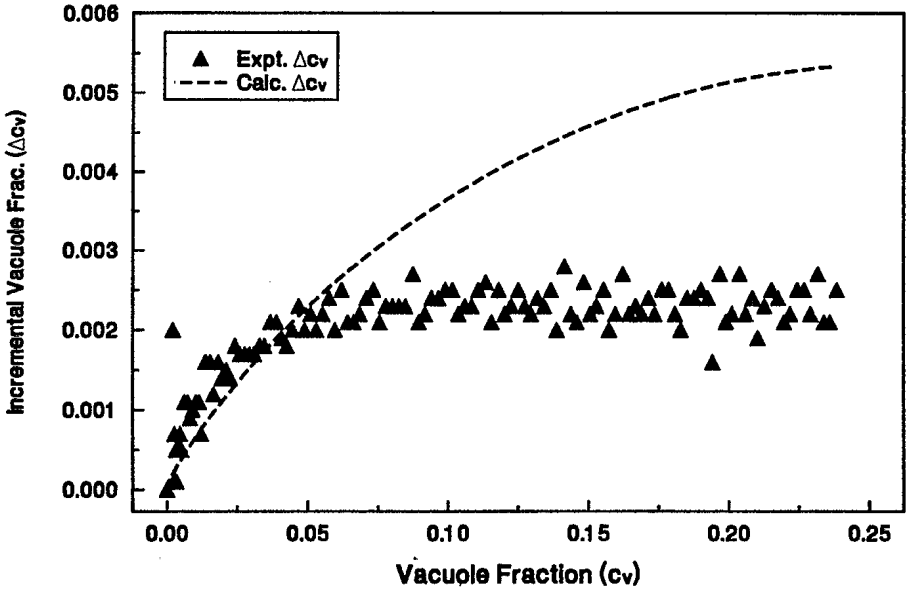

FIGURE 17 – Calculated incremental vacuole concentration behavior for composite T5ML

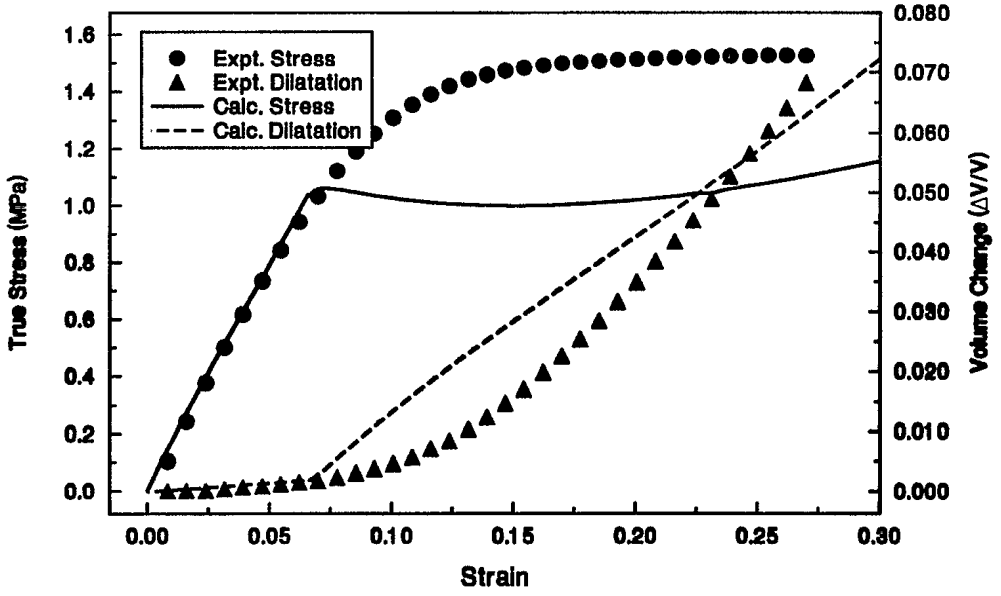

FIGURE 18 – True stress and dilatation behavior calculations for composite T5ML

TABLE IV

Single bead debond results

Specimen	Rate (mm/min)	E_c^{up} (MPa)	E_c^{dn} (MPa)	ϵ_{cr}	G_c^{app} (J/m^2)
1	10	1.47	1.28	0.328	426
2	10	1.68	1.44	0.256	323
3	10	1.71	1.51	0.296	369
4	100	2.08	1.66	0.277	680
5	100	N/A	N/A	N/A	N/A
6	100	1.85	1.52	0.350	835

Notes: E_c^{up} is the average composite modulus when loading and E_c^{dn} is the average modulus when unloading. Results for Specimen 5 are not available because it failed prematurely.

this observation. The average measured adhesion energy G_c^e at 10 mm/min was 374 J/m^2. At 100 mm/min, the average G_c^e was 758 J/m^2. Thus, the relative G_c^e magnitude based on the 10 mm/min result was 2.03. Figure 19 (p. 44) compares the stress-strain data measured for Specimens 3 (10 mm/min) and 6 (100 mm/min). It was interesting to note that the polymer failed around the bead at $\theta \approx 30°$. This was predicted by the finite element results and has been predicted by others (Refs. 9, 70). If the particles in the model composites debonded in a similar manner then this would mean the results in Tables II and III are under estimated by a factor of 2.

It is difficult to pin-point the actual G_c for the different beads used. In general, the treated beads had larger G_c^{app} than the untreated ones. Exceptions to this were the composites containing $c_o^i = 0.3$ of the 31 μm beads. Comparisons of c_v^e versus ϵ_c^e show that the as-received beads debonded earlier than the treated beads in all cases (Fig. 20, p. 45 and Fig. 21). This result demonstrates clearly that the bonding for treated beads was better because more strain energy was required in the composite to induce debonding.

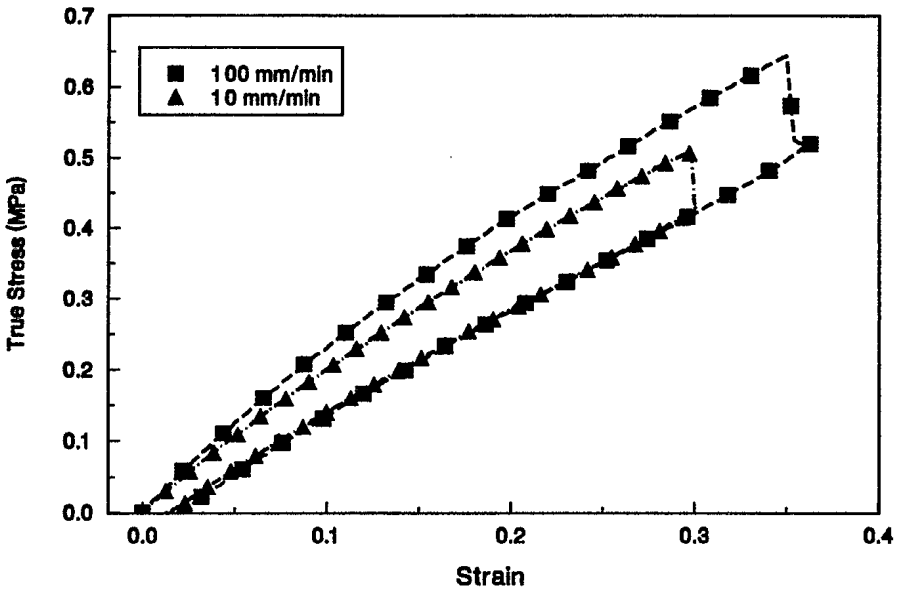

FIGURE 19 – Adhesive energy dissipation for debond specimens tested at 10 mm/min and 100 mm/min

The Tables, however, also suggest that the 130 μm beads have a larger G_c than the 31 μm beads and that G_c increases with increases in c_o^i. There are no obvious reasons why this should be so. The fact that the model can reproduce the $\sigma_c^e - \epsilon_c^e$ and $\Delta V^e/V - \epsilon_c^e$ behavior using G_c^{app} suggests there was validity in assuming equal amounts of surface area have debonded in the actual and model particle distributions. At this point, the values for G_c can only be called "apparent" because the factors which influence this parameter have yet to be precisely determined. This remains a problematic area for this model. Quantification of energy dissipation by mechanisms other than surface creation in composites has been studied in (Refs. 71 - 73) though and may lead the way for further work.

A check was made on the viscoelastic nature of the HTPB used in this study to see whether it could affect the analysis results. A stress relaxation test was carried out on a stick of pure HTPB polymer according to accepted procedures (Refs. 74, 75). The

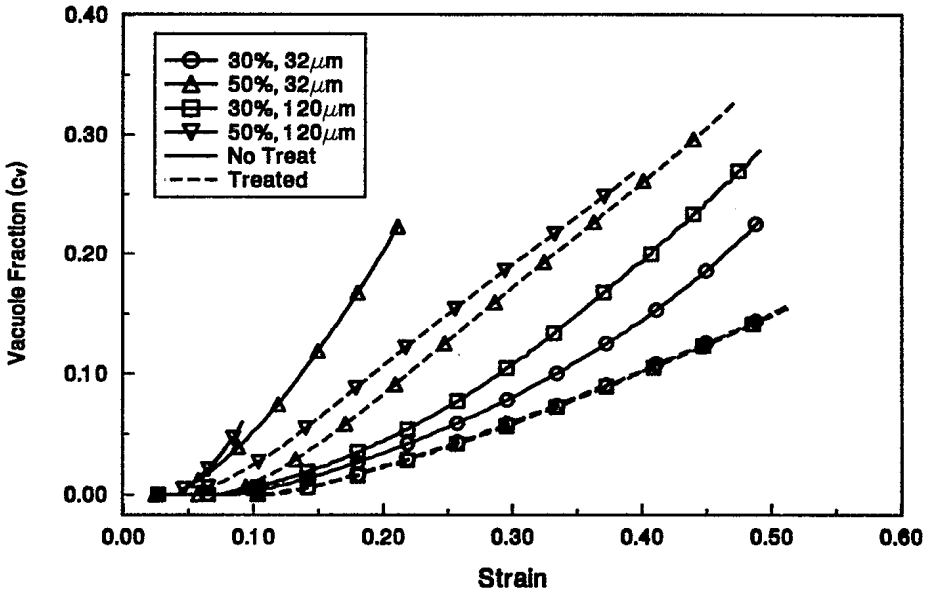

FIGURE 20 – Evolution of vacuole concentration with composite strain for composites tested at 100 mm/min

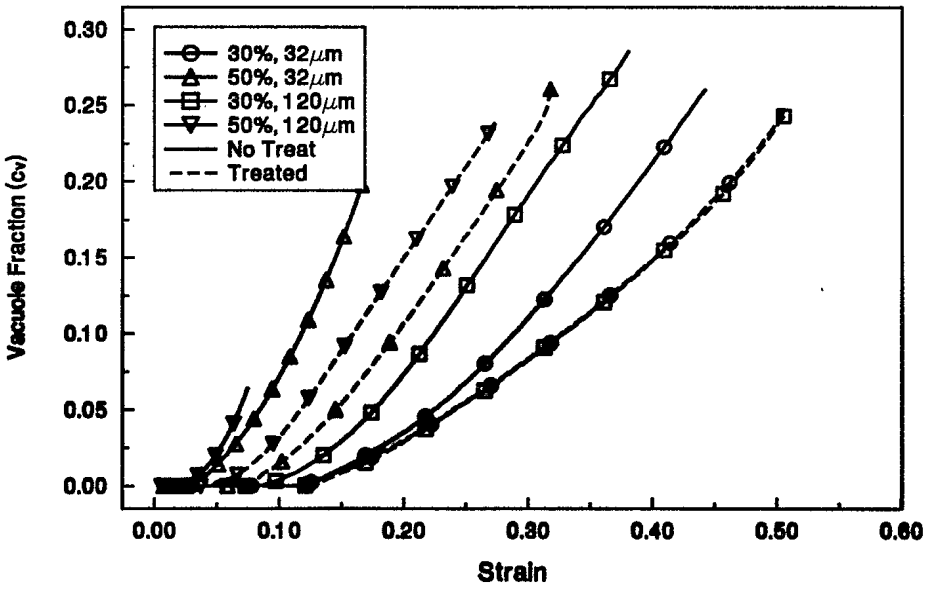

FIGURE 21 – Evolution of vacuole concentration with composite strain for composites tested at 10 mm/min

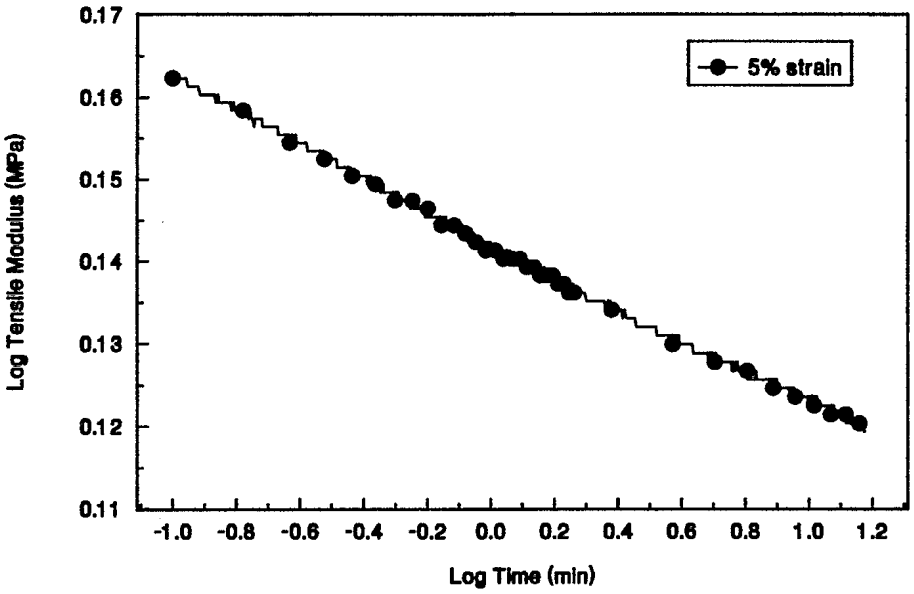

FIGURE 22 – Stress relaxation behavior of HTPB assuming modified power law response

specimen was strained to 5% at 50 mm/min and then allowed to relax for 1000 s. Figure 22 shows the reduced data assuming relaxation follows the modified power law (Ref. 76)

$$E(t) = E_o^r \, t^{-m} \qquad [21]$$

E_o^r is the modulus at $t = 1$ and m is the slope of the data in the log-log plane. The values for E_o^r and m were determined to be 1.387 MPa and 0.0194, respectively. Considering that the duration of a tensile test at 100 mm/min was on the order of 0.5 min. and a test at 10 mm/min was 5 min., the drop in matrix modulus due to time effects is approximately 5%. This is not a significant reduction so the assumption of nonlinear elasticity can be considered justifiable.

5.2 Sensitivity and Predictive Capability of Micro-Mechanical Model

The back-calculated parameters found in the previous section reduced the error in modulus and dilatation to a minimum. The sensitivity of the micromechanical model to changes in the adjustable parameters will be examined from two perspectives. The first is to examine the tendencies the model exhibits with changes in the parameters. This will identify the dominant parameter. The second is to use the model like a user would to predict the mechanical behavior of a composite under development. This will demonstrate the predictive capability of the model given the presence of the adjustable parameters.

Four parameters, namely σ_{cr}, Y_m, F_b and G_c, were needed in the model. Out of these four, only Y_m and F_b can be considered truly adjustable. σ_{cr} was specified by examining the measured stress-strain and dilatation-strain results (Sec. 5.1.1). G_c is a measurable quantity too, although it is not clear how it should be measured. As a consequence, the sensitivity analyses presented in the following sections will use σ_{cr} and G_c^{app} as shown in Tables II and III, p. 31. Y_m and F_b will be varied. Analyses are limited to the 10 mm/min parameters because the 100 mm/min parameters have the same trends based on composite type.

5.2.1 Trends with Y_m and F_b

To make the analysis manageable, the sensitivity of a single point on the $\sigma_c^e - \epsilon_c^e$ and $\Delta V^e/V - \epsilon_c^e$ curves were selected for comparison with the calculated values. These points were called target values and were generally chosen at the mid-way point in the phase where particles were debonding because they would measure the average change in behavior. Table V lists this information along with the corresponding target Y_m and F_b

TABLE V

Target values for sensitivity analyses

Composite	ϵ_c^e	σ_c^e (MPa)	$\Delta V^e/V$	Y_m	F_b $\times 10^{-4}$
T3ML	0.400	1.19	0.023	1.26	1.8
N3ML	0.301	0.626	0.050	0.92	0.24
T5MS	0.249	1.46	0.042	0.83	1.0
N5MS	0.150	0.933	0.028	1.06	1.0

(see also Figs. 10 and 11, p. 37). The term "fractional value" will be used often. This is defined by

$$\text{fraction} = \frac{\text{actual}}{\text{target}} - 1 \qquad [22]$$

The fractional stress and dilatation values were calculated over a fractional Y_m and F_b of ± 0.25 in steps of 0.05. This produced 100 different combinations to examine for each composite. For the sake of brevity, only the graphical results of T5MS and N5MS will be shown. The composites T3ML and N3ML exhibited similar behavior. The FORTRAN listing used for the sensitivity analysis may be found in Appendix D.

A comparison of the fractional stress contours for T5MS (Fig. 23, p. 50) and N5MS (Fig. 25, p. 51) shows that the stress sensitivities are quite different. In T5MS, there are high rates of change centered at fractional $Y_m = -0.20$ and $F_b = 0.15$. In N5MS, the gradient is more uniform. A similar pattern is observed in the dilatational sensitivities (Figs. 24 and 26). In all the Figures, high fractional stresses or low fractional dilatations are seen when fractional F_b is high and Y_m is low. Since a high fractional F_b increases stiffness, one would expect high stresses there. However, a high fractional Y_m, also increases stiffness but low stresses tend to occur in those locations. This shows that the model is influenced more

TABLE VI

Minimum and maximum fractional σ_c and $\Delta V/V$

Composite		Fractional					
		σ_c	F_b	Y_m	$\Delta V/V$	F_b	Y_m
T3ML	max	0.02	0.25	−0.25	0.50	−0.25	0.20
	min	−0.08	−0.25	0.25	−0.30	0.25	−0.25
N3ML	max	0.06	0.25	−0.25	0.65	−0.25	0.20
	min	−0.10	−0.25	0.25	0.05	0.25	−0.25
T5MS	max	0.40	0.15	−0.20	0.40	−0.25	0.25
	min	−0.10	−0.25	0.25	−0.50	0.15	−0.20
N5MS	max	0.08	0.25	−0.25	0.30	−0.25	0.25
	min	−0.08	−0.25	0.25	−0.50	0.25	−0.25

by the value of F_b than Y_m. The Figures reveal that it is better to under estimate F_b and over estimate Y_m if reasonably accurate values of stress (fractional $\sigma_c < 10\%$) are desired. This would be to the detriment of the dilatation results though. Table VI summarizes the locations of the minimum and maximum fractional stresses and dilatations for all composites examined in this section.

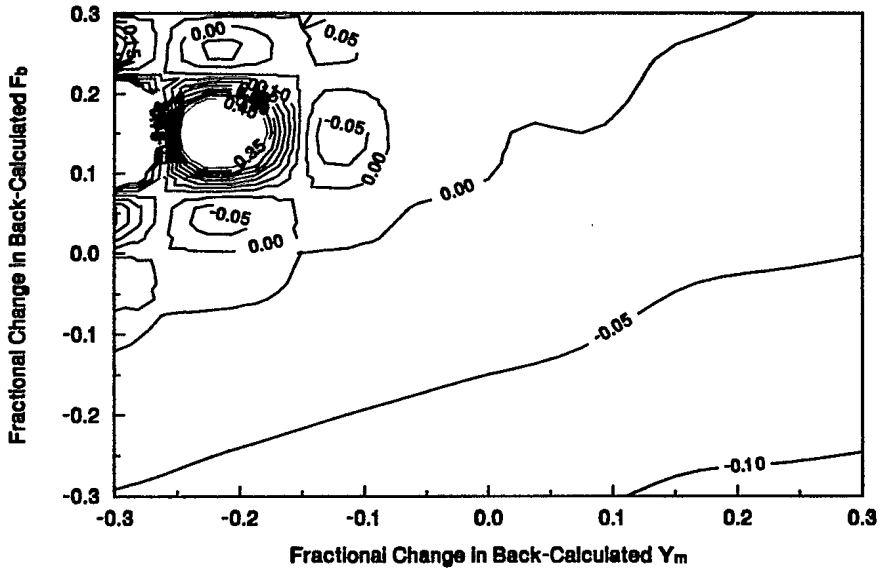

FIGURE 23 – Fractional stress contours for composite T5MS based on variation of interaction factor, Y_m and debonding factor, F_b

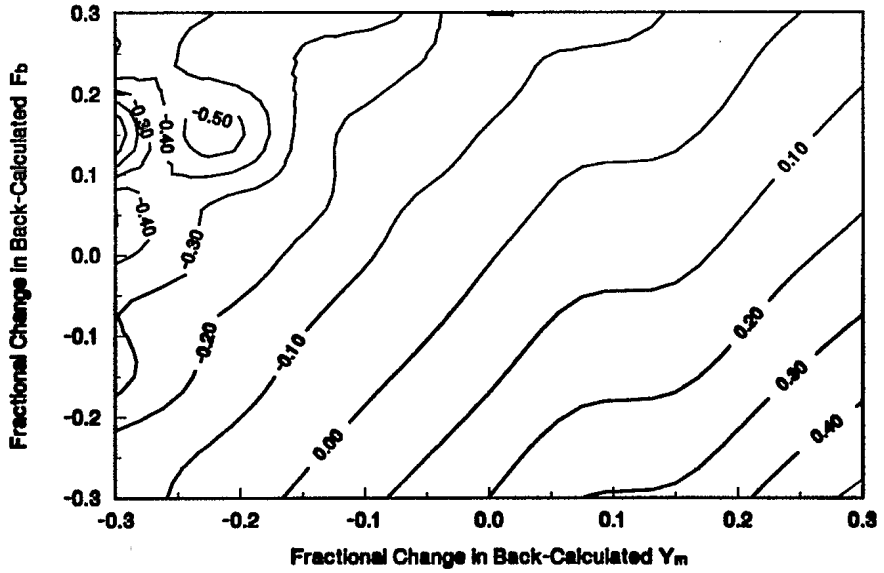

FIGURE 24 – Fractional dilatation contours for composite T5MS based on variation of interaction factor, Y_m and debonding factor, F_b

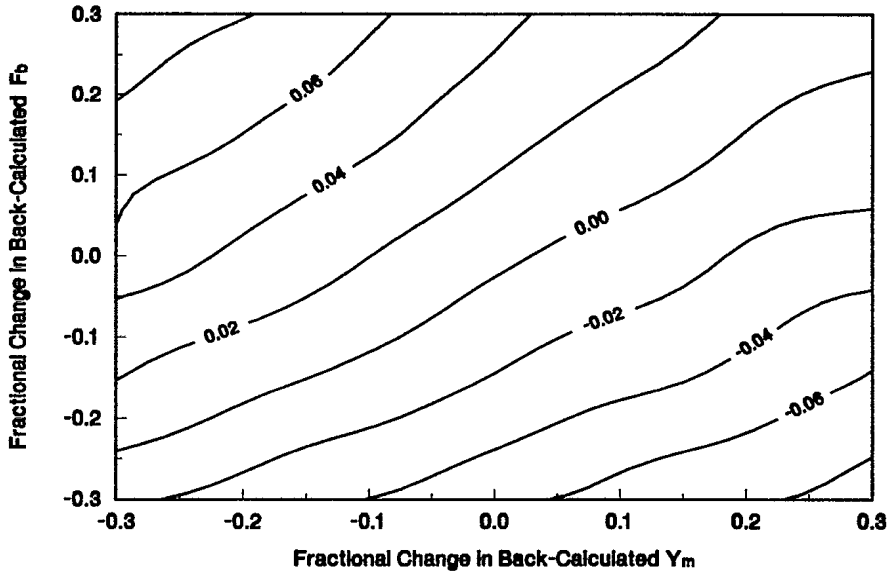

FIGURE 25 – Fractional stress contours for composite N5MS based on variation of interaction factor, Y_m and debonding factor, F_b

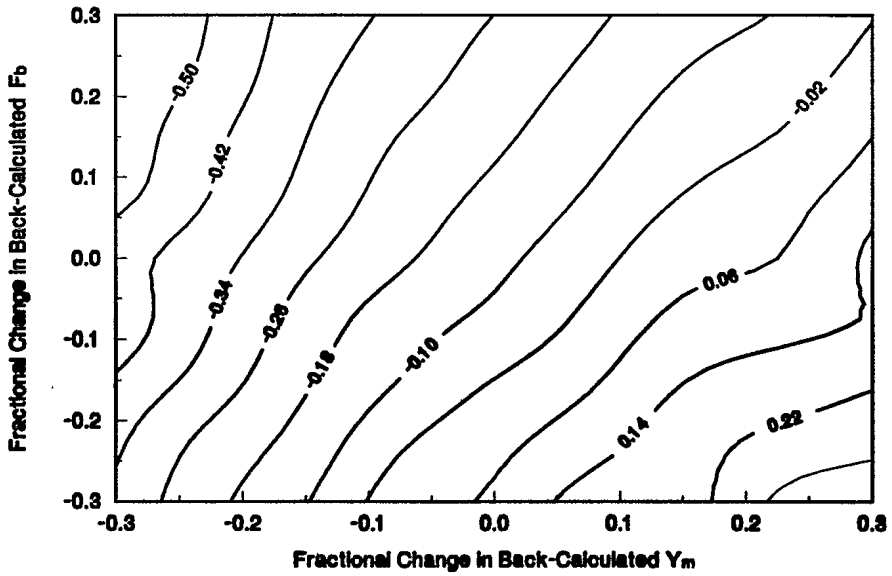

FIGURE 26 – Fractional dilatation contours for composite N5MS based on variation of interaction factor, Y_m and debonding factor F_b

For interest, a similar analysis was carried out for fractional values of σ_{cr} and G_c^{app} while Y_m and F_b were fixed according to Tables II and III. Figures 27 and 28 shows that stress and dilatation are totally dominated by G_c^{app}. For the range studied ($-0.15 <$ frac. $G_c^{app} < +0.15$), the stresses are within 10% of the target values. Dilatation is only slightly affected by changes in G_c^{app} in comparison to the effects of F_b. The analysis was limited to ± 0.15 because greater values caused large increases in fractional stress. The results' insensitivity to variations in σ_{cr} highlight again the importance of having a representative value for G_c.

5.2.2 Predictive Capability of Model

To use the micromechanical model for predicting mechanical behavior of an unknown composite, a set of "best guess" Y_m and F_b values are required. It is evident from Table III that the F_b for treated particles is not in the same range as the F_b for as-received particles. Since it was determined in the previous section that it was better to under estimate F_b, a "best guess" value of $F_b = 1.0 \times 10^{-4}$ was selected for the treated particles and a value of 0.25×10^{-4} was selected for the as-received ones. A reasonable "best guess" value of $Y_m = 1.0$ was selected because the model is less sensitive to variations in Y_m and it was a nice round number. The FORTRAN listing for the micromechanical model is given in Appendix E.

The results of using these "best guess" values are shown in Table VII. For T3ML, N3ML and T5MS, the fractional stresses are less than 0.1 as expected. Composite N5MS fell outside this range because unlike the other untreated composites, its back-calculated F_b was closer to 1.0×10^{-4}. The N5MS values were calculated by hand because they were well beyond the limits of Figs. 25 and 26. The fractional dilatations were high as expected.

TABLE VII

Fractional σ_c and $\Delta V/V$ for best guess Y_m and F_b

Composite	Best Guess		Fractional			
	Y_m	F_b	Y_m	F_b	σ_c	$\Delta V/V$
T3ML	1.0	1.00×10^{-4}	-0.21	-0.44	≈ -0.08	$\approx +0.50$
N3ML	1.0	0.25×10^{-4}	$+0.09$	$+0.04$	≈ -0.02	$\approx +0.30$
T5MS	1.0	1.00×10^{-4}	$+0.20$	$+0.00$	≈ -0.05	$\approx +0.10$
N5MS	1.0	0.25×10^{-4}	-0.06	-3.00	-0.22	$+0.74$

As a final evaluation of these "best guess" values, they were used to predict the behavior of composites T3FS and N3FS. From Fig. 29, p. 55, it can be seen that σ_c^c for T3FS compares well with σ_c^e. It is slightly lower than the σ_c^c calculated with back-calculated values (Fig. 10, p. 37). The $\Delta V^c/V$ is over estimated in relation to $\Delta V^e/V$ and the $\Delta V^c/V$ calculated with back-calculated values (Fig. 11). For N3FS, the trends are similar except the difference between σ_c^c and σ_c^e is more noticeable (Figs. 30 and 10). This shows that it is possible to predict the mechanical behavior of particulate composites if suitable values for σ_{cr} and G_c^{app} are available.

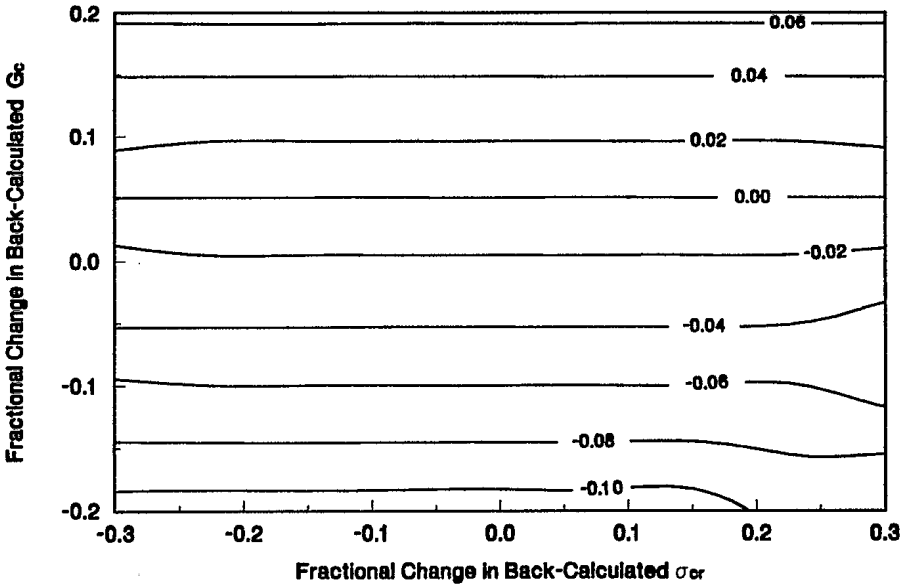

FIGURE 27 – Fractional stress contours for composite T5MS based on variation of critical stress, σ_{cr} and apparent adhesion energy, G_c^{app}

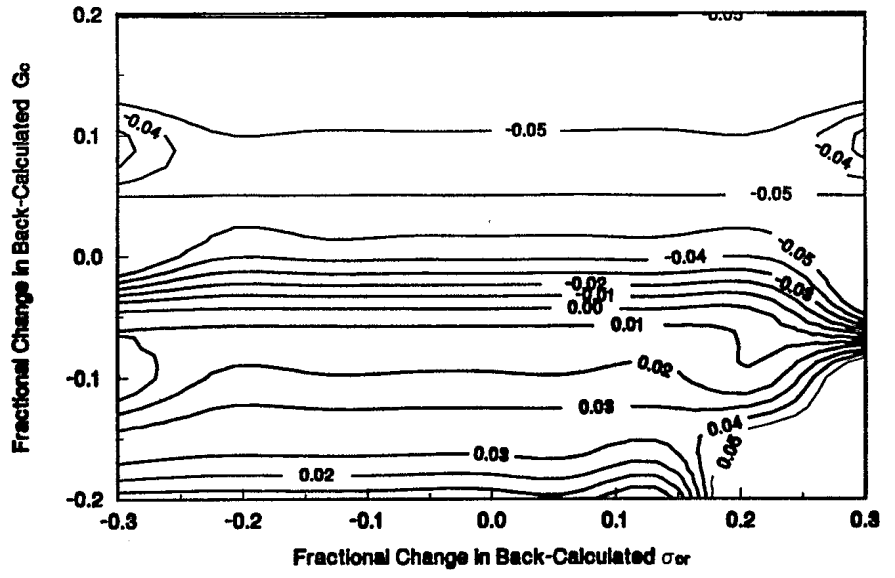

FIGURE 28 – Fractional dilatation contours for composite T5MS based on variation of critical stress, σ_{cr} and apparent adhesion energy, G_c^{app}

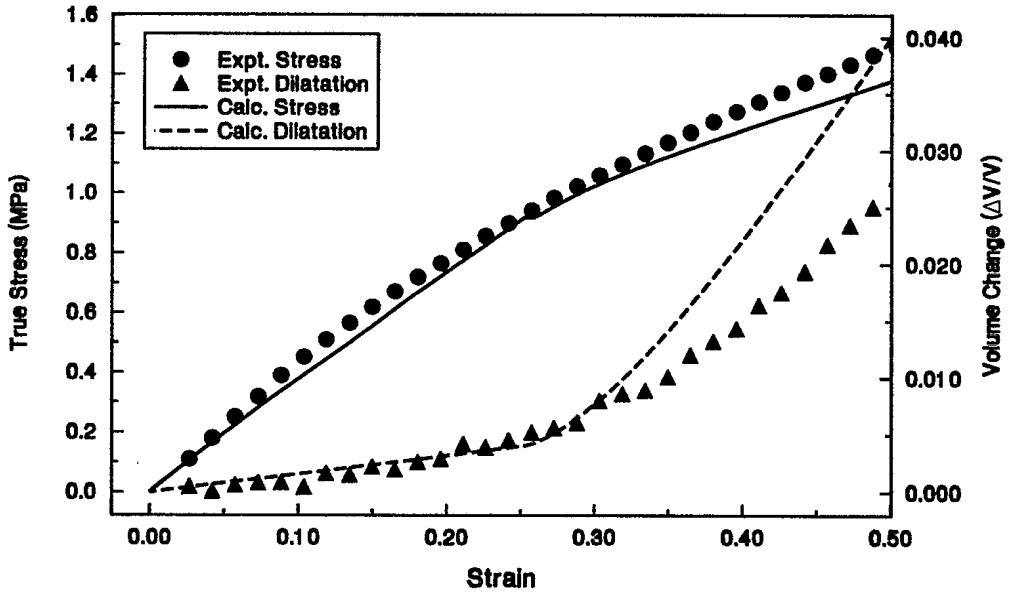

FIGURE 29 – Predicted mechanical behavior of composite T3FS based on best-guess parameters derived from 10 mm/min results

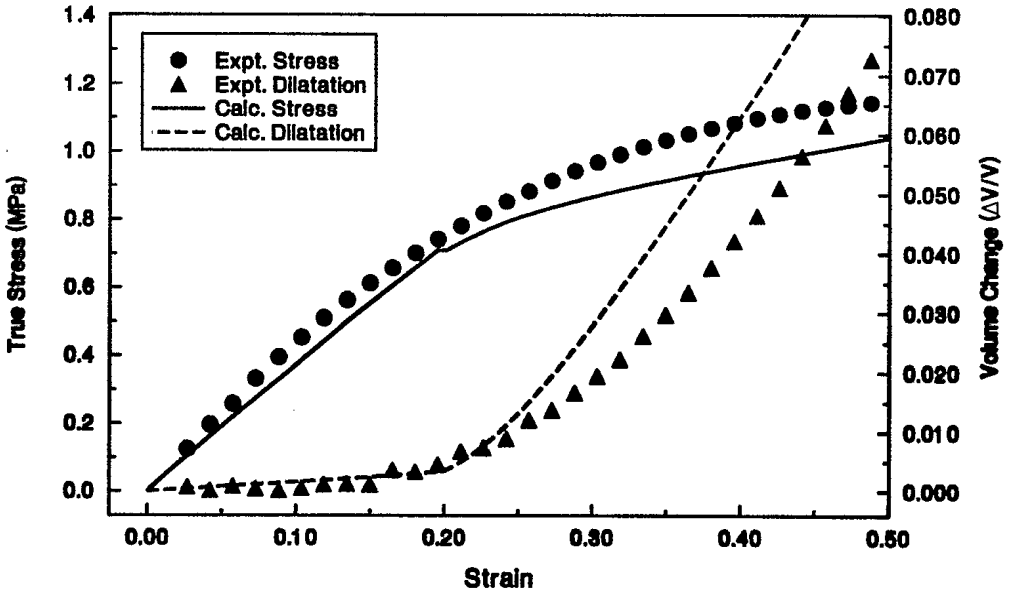

FIGURE 30 – Predicted mechanical behavior for composite N3FS based on best-guess parameters derived from 10 mm/min results

5.2.3 Prediction of Literature Experimental Data

In Ref. 25, the stress-strain behavior of a composite composed of glass bead/HDPE was used to evaluate the Anderson-Farris model. One of the shortcomings of the model was its inability to reproduce the nonlinear behavior at the beginning of the stress-strain curve. To demonstrate the predictive capability of the micromechanical model developed here, predictions for the T25-20 and T100-20 composites were made using the "best guess" F_b and Y_m parameters from the previous section. Critical stress values were set at 25 MPa based upon examination of stress-strain curves. The experimental data did not have volume change measurements because the dilatometer load cell was not capable of withstanding more than a force of 200 lbs. G_c^{app} for each composite had to be found by trial and error. The nonlinear HDPE modulus was characterized using experimental data for pure HDPE up to the measured maximum stress point ($\approx 0.10\epsilon$). Input parameters used for the predictions may be found in Table VIII.

Figures 31 and 32 show that the predicted stress-strain results for T25-20 and T100-20 captured the nonlinear character of the composites before and after the critical stress point. With the G_c^{app}'s selected and the assumption there was no significant reduction in cross-sectional area, the predicted stress at $\epsilon = 0.04$ under estimated the experimental stress by about 15%. This difference could not be reduced through the use of higher G_c^{app}'s because numerical instabilities would appear in the solutions.

In Ref. 57, the behavior of a glass bead/polyurethane composite was studied. Since volume change data was available from Ref. 57, the reduction techniques in Sec. 5.1.1 could be used. The back-calculated Y_m, F_b and G_c^{app} values for the untreated $c_o^i = 0.3$ composite and the treated $c_o^i = 0.3$ and $c_o^i = 0.4$ composites are shown in Table IX.

TABLE VIII

Model parameters for glass bead/HDPE tested at 10 mm/min

Composite	T100-20	T25-20
$\overline{r}$ (μm)	65	15.5
n	0.0374	0.167
c_o^i	0.19	0.22
c_o^v	0.0	0.0
ν_i	0.16	0.16
ν_o	0.34	0.34
G_i (GPa)	30	30
E_o (MPa)	see a	see a
σ_{cr} (MPa)	25	25
Y_m^{bg}	1.0	1.0
$F_b^{bg} \times 10^{-4}$	1.0	1.0
G_c^{app} (J/m^2)	21	17

a $E_o = 965.13 - 11696.5\epsilon + 49450.5\epsilon^2$ (MPa)

$\overline{r}$, average particle radius, n log standard deviation in radius, c_o^i, initial inclusion volume fraction, c_o^v, initial vacuole volume fraction, ν_i, inclusion Poisson ratio, ν_o, matrix Poisson ratio, G_i, inclusion shear modulus, E_o matrix tensile modulus, σ_{cr}, critical stress, Y_m^{bg}, "best-guess" interaction factor multiplier, F_b^{bg}, "best-guess" partial debonding factor, G_c^{app}, apparent adhesion energy.

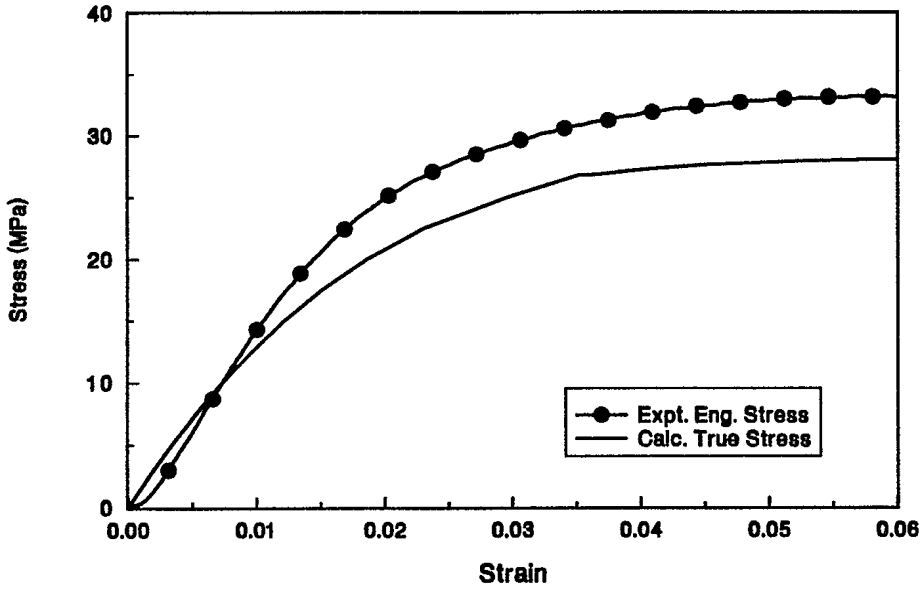

FIGURE 31 – Predicted mechanical behavior for treated glass bead/HDPE composite T25-20 based on best-guess parameters

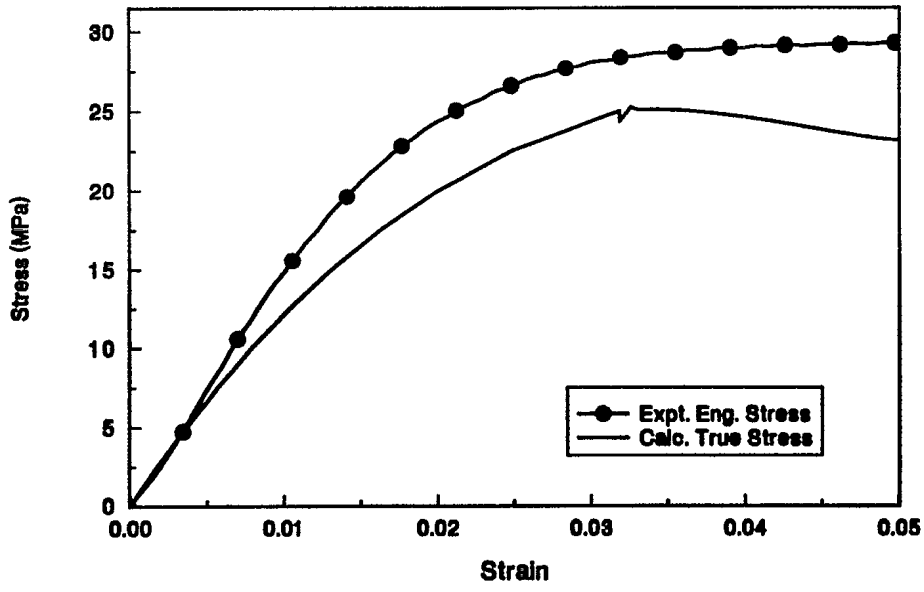

FIGURE 32 – Predicted mechanical behavior for treated glass bead/HDPE composite T100-20 based on best-guess parameters

TABLE IX

Model parameters for glass bead/polyurethane tested at 10 mm/min

Surface Treatment Volume Fraction	Untreated 0.3	Treated 0.3	Treated 0.4
$\bar{r}$ (μm)	12.5	12.5	12.5
n	0.228	0.228	0.228
c_o^i	0.3	0.3	0.4
c_o^v	0.0	0.0	0.0
ν_i	0.16	0.16	0.16
ν_o	0.499	0.499	0.499
G_i (GPa)	30	30	30
E_o (MPa)	see a	see a	see a
σ_{cr} (MPa)	1.5	2.0	1.75
Y_m	2.1	1.81	1.58
$F_b \times 10^{-4}$	1.4	2.7	3.0
G_c^{app} (J/m^2)	9.34	15.0	6.56

a $E_o = 4.30579 - 0.533303\epsilon + 0.227341\epsilon^2$ (MPa)

$\bar{r}$, average particle radius, n log standard deviation in radius, c_o^i, initial inclusion volume fraction, c_o^v, initial vacuole volume fraction, ν_i, inclusion Poisson ratio, ν_o, matrix Poisson ratio, G_i, inclusion shear modulus, E_o matrix tensile modulus, σ_{cr}, critical stress, Y_m, interaction factor multiplier, F_b, partial debonding factor, G_c^{app}, apparent adhesion energy.

In terms of the adhesive characteristics F_b and G_c^{app}, a comparison between the treated and untreated composite parameters show that they are similar to the trends seen for the glass bead/HTPB composites (Tables II and III). When comparing F_b and G_c^{app} for the treated glass bead/polyurethane composites, the $c_o^i = 0.3$ composite has a smaller F_b and a larger G_c^{app} than the $c_o^i = 0.4$ composite. This is opposite to the trends seen for the glass bead/HTPB composites.

Again, "best guess" values for Y_m and F_b were used to predict the mechanical behavior for the selected glass bead/polyurethane materials. For the treated composites, the experimental stress at $\epsilon = 0.3$ was underestimated by approximately 14% (Figs. 33

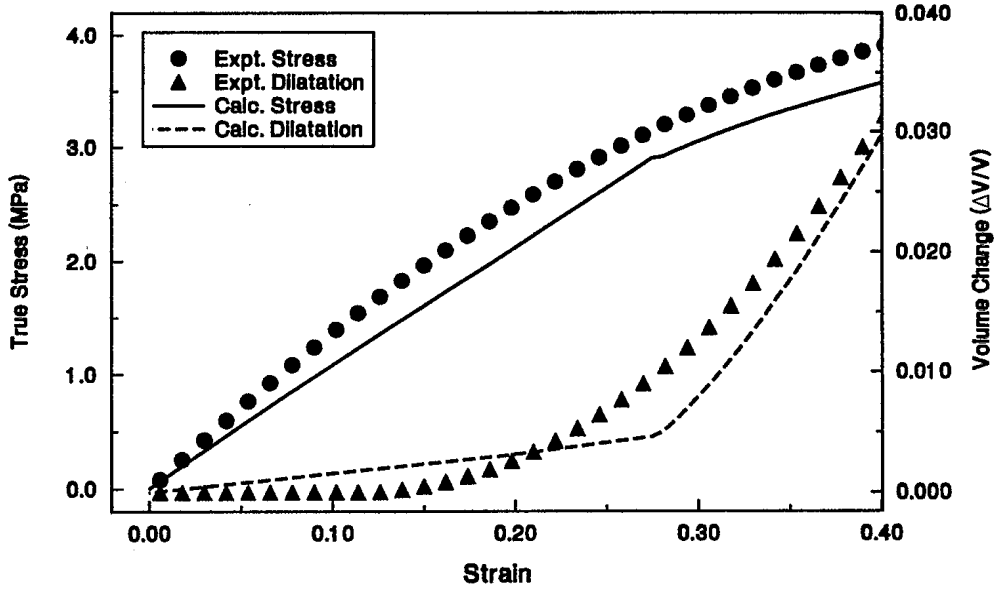

FIGURE 33 – Predicted mechanical behavior for treated glass bead/polyurethane composite based on best-guess parameters, $c_o^i = 0.3$

and 34). For the untreated composite, the stress was underestimated by 24% at $\epsilon = 0.5$ (Fig. 35). Dilatation was overestimated for the untreated $c_o^i = 0.3$ and treated $c_o^i = 0.4$ composites and underestimated for the treated $c_o^i = 0.3$ composite. Given the approximate nature of the F_b's and Y_m's employed in this section, the micromechanical model gave reasonable predictions for the glass bead/HDPE and glass bead/polyurethane composites.

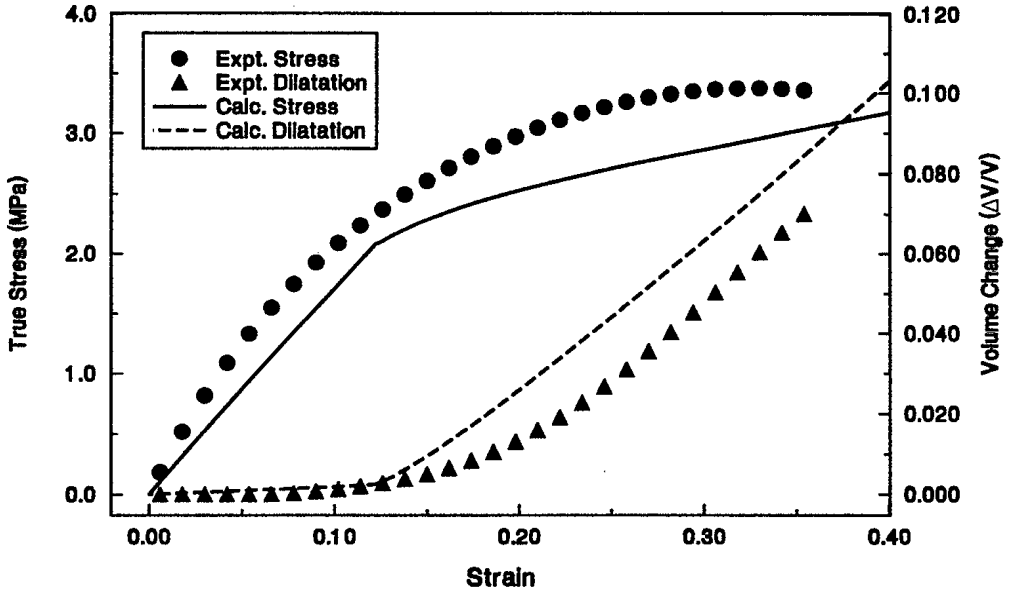

FIGURE 34 – Predicted mechanical behavior for treated glass bead/polyurethane composite based on best-guess parameters, $c_o^i = 0.4$

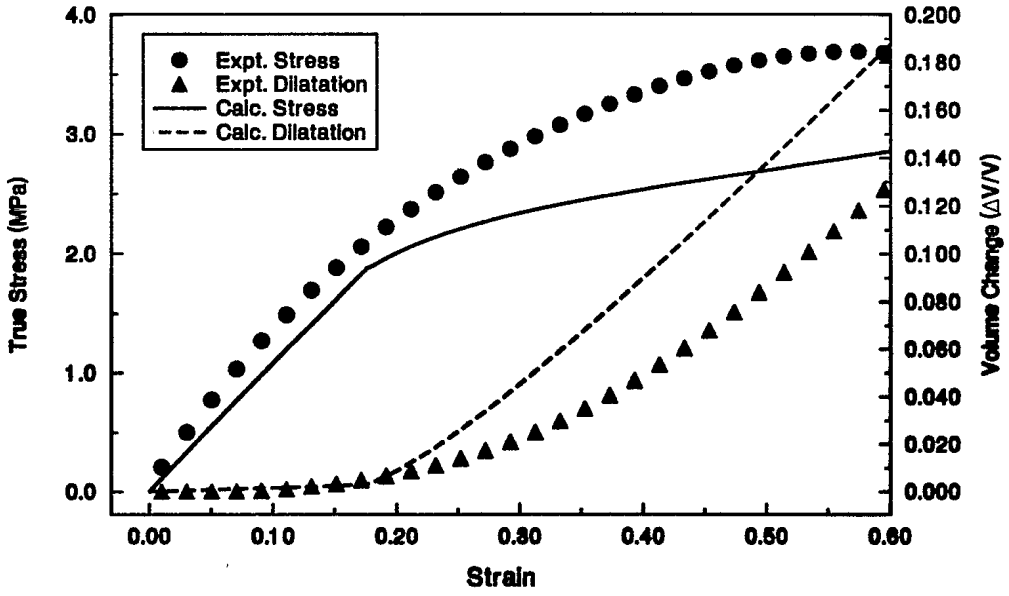

FIGURE 35 – Predicted mechanical behavior for untreated glass bead/polyurethane composite based on best-guess parameters, $c_o^i = 0.3$

6.0 CONCLUSIONS

A micromechanical model for analysis of particulate mechanical behavior has been presented. The assumptions used during the model development were:

1. Nonlinear effects are partly due to matrix and particle debonding.
2. Only well-bonded particles, debonded particles and matrix are present in the composite.
3. Well-bonded particles are characterized by isotropic properties while debonded particles are approximated with orthotropic properties.
4. Matrix is nonlinear elastic.
5. The representative volume element is much larger than the largest particle.
6. The interaction multiplier and debonding factor are constant throughout the loading history.
7. Particles debond progressively from largest sizes to smallest sizes throughout the strain history.

The study was broken into two sections. The first section used the experimental data from a range of glass bead/HTPB composites to back calculate model parameters. Reasonable values for the parameters were found. This showed that the micromechanical model gave a good representation of the processes believed to control mechanical behavior. The deviations between calculated and experimental stress were small when the back-calculated parameters were used. Larger deviations for dilatation existed. These deviations were traced back to differences between the calculated and deduced vacuole volume fractions.

A comparison between the back-calculated interaction multipliers for the different composites suggested that composite modulus is not only a function of volume fraction and particle interaction but it may be influenced by particle size and surface treatment as well. The non-zero debonding factors found in this study suggested that particles cannot be

assumed to fully debond. Assumption 7. was confirmed indirectly by using the incremental vacuole fraction as an indicator of the particle sizes debonded at any given moment.

The second section examined the sensitivity of the model results to small changes in the interaction multiplier and debonding factor. The debonding factor was found to have a dominating effect on the calculated results. Changes in composite stress were less sensitive to changes in this factor than composite dilatation. The results showed it was better to under estimate the debonding factor and to over estimate the interaction multiplier when modeling an unknown composite. An additional analysis showed the apparent adhesion energy also dominated the model results. The model's dependency on representative adhesion characteristics remains a problematic area due to the difficulty in measuring such values.

Based on the sensitivity results, "best guess" interaction and debonding parameters were selected to examine the predictive capability of the model. The critical stress and adhesion energy were assumed known. In most cases, the predicted composite stresses were within 10% of the experimental glass bead/HTPB data. Dilatation was usually over-predicted. As additional tests, the behavior of glass bead/HTPB composites tested at 100 mm/min and the behavior of glass bead/HDPE and glass bead/polyurethane composites were predicted using "best guess" interaction multiplier and debonding factors. The results showed that the model was capable of predicting the mechanical behavior as long as suitable values for critical stress and adhesion energy were available.

7.0 ACKNOWLEDGEMENTS

The author would like to thank the Defence Research Establishment Valcartier management for their support in this work. The assistance given by Mr. P. Carignan and Mr. M. Kervarec, DREV, during the fabrication of the debond specimens is greatly appreciated. The work carried out by Mr. R. Coulombe and Mr. L. Bourret, DREV, to fabricate the glass bead/HTPB composites is also acknowledged.

8.0 REFERENCES

1. Smith, T. L., "Volume Changes and Dewetting in Glass Bead – Polyvinyl Chloride Elastomeric Composites Under Large Deformations", Trans. Soc. Rheol., Vol. 3 (1959) pp. 113-136.

2. Farris, R. J., "The Influence of Vacuole Formation on the Response and Failure of Filled Elastomers", Trans. Soc. Rheol., Vol. 12 (1968) pp. 315-334.

3. Vratsanos-Anderson, L. L. and Farris, R. J., "A Predictive Model for the Mechanical Behaviour of Particulate Composites. Part I: Model Derivation", Polym. Eng. Sci., Vol. 33 (1993) pp. 1458-1465.

4. Vratsanos-Anderson, L. L. and Farris, R. J., "A Predictive Model for the Mechanical Behaviour of Particulate Composites. Part II: Comparison of Model Predictions to Literature Data", Polym. Eng. Sci., Vol. 33 (1993) pp. 1466-1474.

5. Leidner, J. and Woodhams, R. T., "The Strength of Polymeric Composites Containing Spherical Fillers", J. Appl. Polym. Sci., Vol. 18 (1974) p. 1639-1654.

6. Morton, M., Murphy, R. J. and Cheng, T. C., "Role of Filler Modulus and Filler-Rubber Adhesion in Rubber Reinforcement ", Advances in Chemistry Series, Vol. 142 (1974) p. 409-414.

7. Landon, G., Lewis, G. and Boden, J. F., "The Influence of Particle Size on the Tensile Strength of Particulate-Filled Polymers", J. Mater. Sci., Vol. 12 (1977) p. 1605-1613.

8. Nicholson, D. W., "On the Detachment of a Rigid Inclusion from an Elastic Matrix", J. Adhesion, Vol. 10 (1979) p. 255-260.

9. Mitsui, S., Kihara, H., Yoshimi, S. and Okamoto, Y., "Impact Strength Improvement of Polystyrene by Dispersion of Rigid Particulate", Polym. Engng. Sci., Vol. 36 (1996) pp. 2241-2246.

10. Smith, J. C., Kermish, G. A. and Fenstermaker, C. A., "Separation of Filler Particles from the Matrix in a Particulate-Loaded Composite Subjected to Tensile Stress", J. Adhesion, Vol. 4 (1972) p. 109-123.

11. Dreyfuss, P., Gent, A. N. and Williams, J. R., "Tear Strength and Tensile Strength of Model Filled Elastomers", J. Polym. Sci.: Polym. Phys. Ed., Vol. 18 (1980) p. 2135-2142.

12. Pukanszky, B., "Influence of Interface Interaction on the Ultimate Tensile Properties of Polymer Composites", Composites, Vol. 21 (1990) p. 255-262.

13. King, J. A., Buttry, D. A. and Adams, D. F., "Development and Evaluation of Surface Treatments to Enhance the Fiber-Matrix Adhesion in PAN-Based Carbon Fiber/Liquid Crystal Polymer Composites. Pt. 2: Electrochemical Treatments", Polym. Comp., Vol. 14 (1993) p. 292-307.

14. Sumita, M., Ookuma, T., Miyasaka, K. and Ishikawa, K., "Effect of Ultrafine Particles on the Elastic Properties of Oriented Low-Density Polyethylene Composites", J. Appl. Polym. Sci., Vol. 27 (1982) p. 3059-3066.

15. Maiti, S. N. and Mahapatro, P. K., "Mechanical Properties of i-PP/$CaCO_3$ Composites", J. Appl. Polym. Sci., Vol. 42 (1991) p. 3101-3110.

16. Halpin, J. C. and Kardos, J. L., "The Halpin-Tsai Equations: A Review", Polym. Eng. Sci., Vol. 16 (1976) p. 344-352.

17. Fedors, R. F., "Effect of Filler on the Mechanical Behavior of Elastomers", Polymer, Vol. 20 (1979) p. 324-328.

18. Hashin, Z. and Shtrikman, S., "A Variational Approach to the Theory of the Elastic Behaviour of Multiphase Materials", J. Mech. Phys. Solids, Vol. 11 (1963) p. 127-140.

19. Walpole, L. J., "On Bounds for the Overall Elastic Moduli of Composite Materials", J. Mech. Phys. Solids, Vol. 17 (1969) p. 235-251.

20. Hashin, Z., "Extremum Principles for Elastic Heterogenous Media with Imperfect Interfaces and Their Application to Bounding of Effective Moduli", J. Mech. Phys. Solids, Vol. 40 (1992) p. 767-781.

21. Laws, N. and McLaughlin, R., "Self-Consistent Estimates for the Viscoelastic Creep Compliances of Composite Materials", Proc. Royal Soc. London Series A, Vol. 359 (1978) p. 251-273.

22. Lene, F. and Leguillon, D., "Homogenized Constitutive Law for a Partially Cohesive Composite Material", Int. J. Solids Struct., Vol. 18 (1982) p. 443-458.

23. Needleman, A., "A Continuum Model for Void Nucleation by Inclusion Debonding", J. Appl. Mech., Vol. 54 (1987) pp. 525-531.

24. Qiu, Y. P. and Weng, G. J., "A Theory of Plasticity for Porous Materials and Particle-Reinforced Composites", J. Appl. Mech., Vol. 59 (1992) p. 261-268.

25. Wong, F. C. and Ait-Kadi, A. "Mechanical Behaviour of Particulate Composites: Experiments and Micromechanical Predictions", J. Appl. Poly. Sci., Vol. 55 (1995) pp. 263-278.

26. Wong, F. C. and Ait-Kadi, A. "On the Prediction of Mechanical Behavior of Particulate Composites Using an Improved Modulus Degradation Model", to appear in J. Comp. Mater., Vol. 2 (1997).

27. Wong, F. C., "On the Prediction of Mechanical Behavior of Particulate Composites Using an Improved Mori-Tanaka Method", Defence Research Establishment Valcartier, DREV R-9514, UNCLASSIFIED.

28. Mori, T. and K. Tanaka, "Average Stress in Matrix and Average Elastic Energy of Materials with Misfitting Inclusions", Acta Metall., Vol. 21 (1973) pp. 571-574.

29. Ju, J. W. and Chen, T. M., "Effective Elastic Moduli of Two-Phase Composites Containing Randomly Dispersed Spherical Inhomogeneities", Acta Mechanica, Vol. 103 (1994) pp. 123-144.

30. Freudenthal, A. M. and Shinozuka, M., "Shrinkage Stresses in a Thick-Walled Viscoelastic Cylinder Bonded to a Rigid Case", AIAA J., Vol. 1 (1963) pp. 107-115.

31. Schapery, R. A., "Approximate Methods for Thermoviscoelastic Characterization and Analysis of Nonlinear Solid Rocket Grains", ICRPG/AIAA 3rd Solid Propulsion Conf., Paper No. 68-520, June (1968).

32. Farris, R. J., "Development of a Solid Rocket Propellant Nonlinear Constitutive Theory", Report AFRPL-TR-75-20 (1975).

33. Lee, T. Y., "Coupled Thermomechanical Effects in High Solids Propellant", AIAA J., Vol. 17 (1979) pp. 1015-1017.

34. Cost, T. L., "Calculation of Transient Thermal Stresses in Solid Rocket Grains", AIAA/SAE/ASME 18th Conf., Paper No. 82-1098, June (1982).

35. Swanson, S. R. and Christensen and L. W., "A Constitutive Formulation for High-Elongation Propellants", J. Spacecraft, Vol. 20 (1983) pp. 559-566.

36. Buckley, C. P., "Prediction of Stress in a Linear Viscoelastic Solid Strained While Cooling", Rheol Acta, Vol. 27 (1988) pp. 224-229.

37. Burke, M., Woytowitz, P. and Reggi, G., "A Nonlinear Viscoelastic Constitutive Model for Solid Propellant", J. Propulsion and Power, Vol. 8 (1992) pp. 586-591.

38. Ozupek, S. and Becker, E. B., "Constitutive Modeling of High-Elongation Solid Propellant", J. Engng. Mater. Tech., Vol. 114 (1992) pp. 111-115.

39. Ravichandran, G. and Liu, C.T., "Modeling Constitutive Behavior of Particulate Composites Undergoing Damage", Int. J. Solids Structures, Vol. 32 (1995) pp. 979-990.

40. Mochida, T., Minoru, T. and M. Obata, "Effect of Damaged Particles on the Stiffness of a Particle/Metal Matrix Composite", JSME Int. J. Series I, Vol. 34 (1991) pp. 187-193.

41. Weng, G. J., "Some Elastic Properties of Reinforced Solids, with Special Reference to Isotropic Ones Containing Spherical Inclusions", Int. J. Engng. Sci., Vol. 22 (1984) pp. 845-856.

42. Tandon, G.P. and G.J. Weng, "A Theory of Particle-Reinforced Plasticity", Trans. ASME, Vol. 55 (1988) pp. 126-135.

43. Tszeng, T.C., "Micromechanics Characterization of Unidirectional Composites during Multiaxial Plastic Deformation", J. Comp. Mater., Vol. 28 (1994) pp. 800-820.

44. Liming, Y. and Zhaoxiang, Z., "A Micromechanical Analysis of the Nonlinear Elastic and Viscoelastic Constitutive Relation of a Polymer Filled with Rigid Particles", Acta Mech. Sinica, Vol. 10 (1994) pp. 176-185.

45. Ju, J.W. and Chen, T-M., "Micromechanics and Effective Elastoplastic Behavior of Two-Phase Metal Matrix Composites", Trans. ASME, Vol. 116 (1994) pp. 310-318.

46. Ju, J. W. and Chen, T. M., "Effective Elastic Moduli of Two-Phase Composites Containing Randomly Dispersed Ellipsoidal Inhomogeneities", Acta Mechanica, Vol. 103 (1994) pp. 103-121.

47. Chen, C-H., Wu, C-H., Wang, Y-C. and Cheng, C-H., "Micromechanics of Nonlinear Behavior in Solid-Filled Mooney-Rivlin Rubber Specimen", J. Propulsion Power, Vol. 12 (1996) pp. 991-994.

48. Chen, C-H. and Cheng, C-H., "Secant Moduli of a Glass Bead-Reinforced Silicone Rubber Specimen", J. Comp. Mater., Vol. 30 (1996) pp. 69-83.

49. Favier, S., Canova, G.R. and Baudelet, B., "A Micromechanical Incremental Modelling of Damage in Materials Containing Second Phase Particles", Acta Metall. Mater., Vol. 43 (1995) pp. 1249-1258.

50. Zhao, Y.H. and Weng, G.J., "A Theory of Inclusion Debonding and Its Influence on the Stress-Strain Relations of a Ductile Matrix Composite", Int. J. Damage Mech., Vol. 4 (1995) pp. 196-211.

51. Yamada, Y., Kawasaki, A., Taya, M. and Watanabe, R., "Effect of Debonding at the Phase Interface on Young's Modulus of Sintered PSZ/Stainless Steel Composite", Mater. Trans., JIM, Vol. 35 (1994) pp. 814-820.

52. Wong, F. C., "Mechanical Behaviour of Particulate Composites: Experiments and Micromechanical Predictions", Defence Research Establishment Valcartier, Report No. R-9403/94 (1994).

53. Hoffman, O. and Sachs, G., "Introduction to the Theory of Plasticity for Engineers", McGraw-Hill Book Co. Inc., New York, NY (1953).

54. Mase, G. E. and Mase, G. T., "Continuum Mechanics for Engineers", CRC Press, Boca Raton, FL (1991).

55. McCullough, R. L., "Micro-Models for Composite Materials - Particulate and Discontinuous Fiber Composites - Vol. 2", Delaware Composites Design Encyclopedia, (1990) pp. 93-142.

56. Bennett, J. G. and Haberman, K. S., "An Alternate Unified Approach to the Micromechanical Analysis of Composite Materials", J. Comp. Mater., Vol. 30 (1996) pp. 1732-1747.

57. Yilmazer, U. and R. J. Farris, "Mechanical Behaviour and Dilatation of Particulate-Filled Thermosets in the Rubbery State", J. Appl. Polym. Sci., Vol. 28 (1983) pp. 3369-3386.

58. Cook, R. D., Malkus, D. S. and Plesha, M. E., "Concepts and Applications of Finite Element Analysis", John Wiley and Sons, New York, NY (1989) pp. 502-504.

59. "Uniaxial Tensile Tests at Constant Strain Rate", Solid Propellant Mechanical Behaviour Manual, Chemical Propulsion Information Agency, Publ. 21 (1970) p. 4.3.2-1.

60. Anderson, L. L., "A Predictive Model for the Mechanical Behaviour of Particulate Composites", Ph. D. Dissertation, Univ. of Massachusetts (1989).

61. Test Method for Tensile Properties of Plastics, ASTM D638-82a (1984).

62. Density and Specific Gravity (Relative Density) of Plastics by Displacement, ASTM D792-86 (1987).

63. Lipatov, Y.S., Babich, V.F. and Todosijchuk, T.T., "Determination of Strength of Adhesion Contacts in Filled Elastomers from Elastic Modulus Change", J. Adhesion, Vol. 35 (1991) pp. 187-202.

64. Hamed, G. R., "Energy Dissipation and the Fracture of Rubber Vulcanizates", Rubber Chem. Tech., Vol. 64 (1991) pp. 493-500.

65. Herrera-Franco, P. and Drzal, L. T., "Comparison of Methods for the Measurement of the Fibre/Matrix Adhesion in Composites", Composites, Vol. 23 (1992) pp. 2-27.

66. Mower, T. M. and Argon, A. S., "An Experimental Technique to Measure the Adhesive Strength Between Inclusions and Transparent Matrices", J. Mater. Sci., Vol. 31 (1996) pp. 1585-1594.

67. Everett, R. K., "Quantification of Random Fiber Arrangements Using a Radial Distribution Function Approach", J. Comp. Mater., Vol. 30, (1996) pp. 748-758.

68. Tong, W. and Ravichandran, G., "Effective Elastic Moduli and Characterization of a Particulate Metal-Matrix Composite with Damaged Particles", Composite Sci. Tech., Vol. 52 (1994) pp. 247-252.

69. Bazhenov, S., "The Effect of Particles on Failure Modes of Filled Polymers", Polym. Engng. Sci., Vol. 35 (1995) pp. 813-822.

70. Tandon, G. P. and G. J. Weng, "Stress Distribution In and Around Spheroidal Inclusions and Voids at Finite Concentration", J. Appl. Mech., Vol. 53 (1986) pp. 511-518.

71. Friedreich, K., Karsh, U.A., "Failure Processes in Particulate Filled Polypropylene", J. Mater. Sci., Vol. 16 (1981) pp. 2167-2175.

72. Needleman, A., "Micromechanical Modelling of Interfacial Decohesion", Ultramicroscopy, Vol. 40 (1992) pp. 203-214.

73. Huang, Y., Hunston, D.L. et.al., "Mechanisms of Toughening Thermoset Resins in Toughened Plastics I", Advances in Chemistry Series, ACS, No. 233 (1993) pp. 1-35.

74. "Viscoelastic Property Tests", Solid Propellant Mechanical Behaviour Manual, Chemical Propulsion Information Agency, Publ. 21 (1970) p. 4.3.6-1.

75. "Explosive Materials, Physical/Mechanical Properties – Stress Relaxation Test in Tension (DRAFT)", NATO STANAG 4507, Sept. (1995).

76. Lai, J. and Bakker, A., "An Integral Constitutive Equation for Nonlinear Plasto-Viscoelastic Behavior of High-Density Polyethylene", Polym. Engng. Sci., Vol. 35 (1995) pp. 1339-1347.

APPENDIX A

ANSYS INPUT FILE FOR DEBOND SPECIMEN ANALYSIS

```
/COM, ANSYS REVISION 5.0A
/COM, REVISED INCLUSION INTERPHASE MODEL EXPTL SET-UP
! MODEL GEOMETRY PARAMETERS
HHGT = 1.25          ! HOLDER HEIGHT (IN)
PRAD = 0.486         ! PARTICLE RADIUS (IN)
CRAD = 1.025         ! CYLINDER RADIUS (IN)
CHGT = 1.5           ! CYLINDER HEIGHT (IN)
CHGT = 0.5*CHGT      ! REDUCE TO 1/2 HEIGHT FOR SYMMETRY
ITHK = 0.025         ! INTERPHASE THICKNESS (IN)
RSRG = 0.250         ! SHOULDER RADIUS (IN)
SHGT = 0.050         ! SHOULDER HEIGHT (IN)
GHGT = RSRG+SHGT     ! GROOVE HEIGHT (IN)
EPROP = 15          ! MATRIX MODULUS (PSI)
TOL = 0.0001         ! DIMENSIONAL TOLERANCE (IN)
TTOL = 1             ! ANGULAR TOLERANCE (DEGREES)
! LOADING PARAMETERS
STRN = 0.2           ! APPROXIMATE APPLIED STRAIN
EFAC = 1.000        ! 1=PERFECT BOND, 0.0001=NO BOND
BSTP = 4             ! BEGINNING LOAD STEP
LSTP = 4             ! LAST LOAD STEP TO CALCULATE MAX=4
DISP = -STRN*CHGT    ! APPLIED DISPLACEMENT (IN)
! SET UP DATABASE
/FILNAM,SYM-2VFB
! LINE DIVISIONS
NANG = 5             ! ANGULAR DEG, PARTICLE/INTERPHASE/GLUE
NSHT = 28            ! LINEAR DIV, CYLINDER HEIGHT SHORT SIDE
NLNG = 18            ! LINEAR DIV, CYLINDER HEIGHT LONG SIDE
NBED = 10            ! LINEAR DIV, BEAD RADIUS
NRSH = 10            ! LINEAR DIV, SHORT SIDE CYL RADIUS
NRLG = 10            ! LINEAR DIV, CYLINDER RADIUS
NHSH = 8             ! LINEAR DIV, SHORT SIDE HOLDER HEIGHT
NHLG = 15            ! LINEAR DIV, HOLDER HEIGHT
NINT = 2             ! LINEAR DIV, INTERPHASE
```

```
NSHL = 4              ! LINEAR DIV, SHOULDER
NANS = 10             ! ANGULAR DEG, SHOULDER RADIUS
! MATERIAL PROPERTIES
EMAT = EPROP          ! MATRIX MODULUS (PSI)
VMAT = 0.49           ! MATRIX POISSON RATIO
AMAT = 10E-5          ! MATRIX THERMAL CO-EFF (/F)
EINT = EMAT*EFAC      ! INTERPHASE
VINT = 0.49
AINT = 10E-5
EGLU = 0.3E6          ! GLUE
VGLU = 0.35
AGLU = 4.5E-5
EHLD = 10.6E6         ! HOLDER
VHLD = 0.33
AHLD = 12.9E-6
EBED = 11E6           ! BEAD
VBED = 0.23
ABED = 5.1E-6
/TITLE, SINGLE SPHERE EMBEDDED IN MATRIX W/INTERPHASE, SYMMETRIC
/PREP7
/COM, 8-NODED QUAD, AXISYMMETRIC
ET,1,82,,,1
/COM, MATERIAL PROPERTIES
MP,EX,1,EMAT
MP,NUXY,1,VMAT
MP,ALPX,1,AMAT
MP,EX,2,EINT
MP,NUXY,2,VINT
MP,ALPX,2,AINT
MP,EX,3,EGLU
MP,NUXY,3,VGLU
MP,ALPX,3,AGLU
MP,EX,4,EHLD
MP,NUXY,4,VHLD
MP,ALPX,4,AHLD
MP,EX,5,EBED
MP,NUXY,5,VBED
```

```
MP,ALPX,5,ABED
/COM, KEYPOINT ID
K,1,0,0
K,2,PRAD,0
K,3,0,PRAD
K,4,0,CHGT
K,5,CRAD-RSRG,CHGT
K,6,CRAD,0
K,7,0,PRAD+ITHK
K,8,PRAD+ITHK,0
K,9,0,-PRAD
K,10,0,-(PRAD+ITHK)
K,11,0,-HHGT
K,12,CRAD,-HHGT
K,13,CRAD,SHGT
K,14,CRAD-RSRG,GHGT
K,15,CRAD,GHGT
/COM, LINE DEFINITION
L,1,2
L,1,3
LARC,2,3,1,PRAD
L,7,4
L,4,5                    ! L5
L,14,5
L,6,8
LARC,8,7,1,PRAD+ITHK
L,2,8
L,3,7                    ! L10
L,1,9
LARC,9,2,1,PRAD
LARC,10,8,1,PRAD+ITHK
L,9,10
L,10,11                  ! L15
L,11,12
L,12,6
L,6,13
LARC,13,14,15,RSRG
```

```
/COM, AREA GENERATION
A,8,6,13,14,5,4,7
A,3,2,8,7
A,9,10,8,2
A,1,2,3
A,1,9,2
AL,13,15,16,17,7
/COM, LINE SEGMENTATION
LESIZE,1,,,NBED,0.5
LESIZE,2,,,NBED,0.5
LESIZE,3,,NANG
LESIZE,4,,,NSHT,3
LESIZE,5,,,NRLG
LESIZE,6,,,NLNG
LESIZE,7,,,NRSH,-3
LESIZE,8,,NANG
LESIZE,9,,,NINT
LESIZE,10,,,NINT
LESIZE,11,,,NBED,0.5
LESIZE,12,,NANG
LESIZE,13,,NANG
LESIZE,14,,,NINT
LESIZE,15,,,NHSH
LESIZE,16,,,NRLG
LESIZE,17,,,NHLG,0.5
LESIZE,18,,,NSHL
LESIZE,19,,,NANS
/COM, MESH GENERATION
TYPE,1
MAT,1
AMESH,1
MAT,2
AMESH,2
MAT,3
AMESH,3
MAT,4
AMESH,6
```

```
MAT,5
AMESH,4,5
FINI
/SOLU
DL,5,1,SYMM
DL,4,1,SYMM
DL,10,2,SYMM
DL,2,4,SYMM
DL,11,5,SYMM
DL,14,3,SYMM
DL,15,6,SYMM
/PBC,U,,1
TIMINT,OFF
TIME,1
DK,11,UY,0.25*DISP,,1
DK,12,UY,0.25*DISP,,1
KBC,1
LSWRITE
TIME,2
DK,11,UY,0.5*DISP,,1
DK,12,UY,0.5*DISP,,1
KBC,1
LSWRITE
TIME,3
DK,11,UY,0.75*DISP,,1
DK,12,UY,0.75*DISP,,1
KBC,1
LSWRITE
TIME,4
DK,11,UY,1.0*DISP,,1
DK,12,UY,1.0*DISP,,1
KBC,1
LSWRITE
LSSOLVE,BSTP,LSTP
FINI
SAVE
/POST1
```

```
SET,1
NSEL,S,LOC,Y,-HHGT-TOL,-HHGT+TOL
FSUM
ESEL,S,MAT,,1,2
NSLE,S
!/SHOW,SYM-2VF,GRP
!PLNSOL,S,1
!PLNSOL,EPEL,Y
!/SHOW,HALO
ETABLE,SENE,SENE
SSUM
```

APPENDIX B

FORTRAN PROGRAMS FOR BACK-CALCULATION OF MODEL PARAMETERS

Analysis Procedure

1. Set $F_b = 0$, $Y_m = 1$, $G_c = 1$. Set step sizes and % err to values shown in Sample Input File. Note % err here means fractional error and not percentage error.

2. Select σ_{cr} from combined stress-dilatation-strain data. Choose point on true stress curve where it starts to separate from the engineering stress curve.

3. Enter in the rest of the required parameters for the input file and run P15.FOR.

4. Program will find Y_m that reduces error in E_c. It may or may not be able to carry out the F_b optimization.

5. If P15.FOR cannot do the F_b optimization, set "Y_m max. iter." to 1 and increase Y_m until the program can start iterating to find F_b. To be consistent find the lowest Y_m that allows this to happen.

6. Allow program to find optimal F_b. If "frac. rem. c_i^o" falls below 0.05, note F_b where this takes place and force program to that F_b setting by using it in the input file and setting F_b iter. to 1.

7. Check dV/V and σ fit. Usually dV/V fit is stiffer than the experimental results. σ fit is generally very good.

8. Look at the $\frac{G_c dA}{V_o dc}$ and $\sum \frac{G_c dA}{V_o dc}$ results in the "r" file. If the strain energy jumps up by a large amount then slowly increases instead of smoothly increasing, the σ_{cr} must be increased. Sometimes this feature is not always possible to get rid of. Try to find a σ_{cr} that will give the smallest step in those cases. Once σ_{cr} is changed then steps 3 to 7 have to be repeated.

9. Note % err for Y_m and F_b if they are not the default values.

10. Re-run P15.FOR then run P14-DIST.FOR.

11. Select NTOT so that $dC_v^{cal} \approx dC_v^{exp}$.

12. Note G_c^{cal} and enter it into the input file. Re-run P14-DIST.FOR.

13. G_c in output should be approx. 1.0. Note avg. M_G and its standard deviation. Large values of M_G means fit is bad between the assumed particle distribution debonding and the actual debonding.

14. Note ISKIP value and debond angle used in P14-DIST.FOR. Input values into "-14" input file.

15. Run P14.FOR to produce fitted stress-dilatation-strain results.

16. Make changes to ISKIP and debond angle if necessary. Ensure that the ISKIP value corresponds to the data interval in the output file.

17. Re-run P14.FOR if necessary.

Sample P15 and P14-DIST Input File

```
ct3f1, part intact, vacuole          ! information header
220                                  ! no. pts. in curve (NOT USED)
1                                    ! no. distributions
65.0                                 ! avg. particle radius (micron)
0.0374                               ! log std. dev.
0.311,0.0                            ! inclusion and void fraction
4609                                 ! sample volume (mm3) (NOT USED)
2.1E-4,1.69,0,1,0                    ! Fb,Ym,w-type,m-type,v-type
0.4363E6,30E9                        ! matrix and inclusion shear modulus (Pa)
0.495,0.16                           ! matrix and inclusion Poisson ratio
30E9,34.12E9                         ! matrix and inclusion bulk modulus (Pa)
5.720,0                              ! Gc, pressure (NOT USED)
2,1E-3,20                            ! a-type, conv. tol., max. iterations
10,0.5E6,0.0,2                     !no.steps init.,crit.strs,dbnd angle,avg.fac.
1.522255E6,-0.460286E6,0.270235E6    ! nonlinear co-effs in Young's modulus (Pa)
0.1,1,0.008,4                        ! Ym step,Ym max iter,Ym %err,start index
0.0001,0.005,1                       ! Cv step,Cv %err,start index
0.1E-4,1,0.001                       ! Fb step,Fb max iter,Fb abs.err
```

FORTRAN Listing for P15.FOR

```
C==== main program
C      P15.FOR
C====
C     program uses experimental stress-strain data to deduce optimal
C     particle interaction Ym and debond fraction Fb. The void fraction
C     is calculated on assumption that Ym and Fb are average values for
C     entire range of composite strain. built using subroutines written
C     from P14.FOR.
C
C     user enters the following information:
C        number of points (not used)
C        number of particle distributions
C        avg radius and std dev of each distribution
C        volume fraction of filler and voids of each distribution
C        sample volume (not used)
C        fraction debond (Fb), YMULT, w-type, m-type, v-type
C        matrix and filler shear modulus (use matrix LE value)
C        matrix and filler poisson ratio
C        void shear and bulk moduli (if both values zero model as voids,
C          if non-zero, use filler or pseudo-filler values)
C        adhesion energy and applied pressure (not used)
C        a-type, matx.conv.tol.(not used), mat.strn.max.iter.(not used)
C        # pts (not used), crit.stress, debond angle(n/u) and iskip(n/u)
C        coefficients k0, k1, k2 for fitted matrix Young's modulus
C        Ym step,Ym max iter,Ym %err,start index
C        Cv step,Cv %err,start index
C        Fb step,Fb max iter,Fb abs.err
C
C     %err really means fractional error and not percentage error.
C
C     information may be entered using keyboard or by input data file.
C     implementation of Mori-Tanaka solution extended for 3-phase and
C     particle interaction. constituent material properties
C     designated as follows: 1-inclusion,2-void or vacuole,3-matrix.
C     fraction debond (FDBND) for orthotropic properties in loading
```

```
C     direction, multiplier for rad. dist. func. (YMULT), w-type
C     designates use inclusion or void properties in
C     calc of Wv matrix (0=void, 1=inclusion), m-type determines type of
C     particle interaction used (0=none,1=inclusion, 2=inclusion and
C     void or vacuole), v-type determines isotropic or orthotropic matl
C     (0=orthotropic,1=isotropic), a-type determines if strain
C     augmentation is used (0=none, 1=yes in initial only, 2=yes in
C     initial and debond sections)
C
C     to write out intermediate data like modulus and Poisson data, set
C     NDIST = -NDIST
C
C     implementation of strain-dependent matrix modulus. added COMMON
C     block /PROPC/, rearranged conditional statement calling sbr
C     CALCIO, CALCCV and CMPRP0. included composite strain in the MTPRP
C     and CALCIO variable lists. stored augmented strain in array STRNP1
C     in /PROPC/.
C
C     take out offset for ECMPX calcs in SBR YMFIND and CVFIND, this
C     causes modulus values to be under-estimated when first few stress
C     values are not near zero. Also corrected stress and dilatation
C     calculations to use previous ECMP and POISC values for the current
C     strain level. this corresponds with what is done in P14.
C
C     to compile and link: fl pxx.for graphics.lib. the files
C     MSGRAPH.FOR and GRFDEF.FOR should be in the same directory unless
C     a temporary variable has been set up to point to the location of
C     include files. these files contain graphics routines necessary to
C     plot stress-strain curve on screen.
C
C     added statement to read in debond angle from input file.
C
C     last revision: 23 OCT 1996  16h30
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
```

```
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 250,NPTMX = 250,NTDIS = 1)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /MATRA/ BETA(2),WI(3,3),WV(3,3),IDENT(3,3)
      COMMON /MATRB/ S(3,3),CA(3,3),CB(3,3),CE(3,3),CF(3,3)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /PROPD/ YMPARM(4),CVPARM(3),FBPARM(3)
      COMMON /VERIF/ VSTRN(NPTMX),VSTRS(NPTMX),VDVV(NPTMX),
     *   VTSTRS(NPTMX),EDIFF(NPTMX),VDIFF(NPTMX),VPOIS(NPTMX),
     *   PDIFF(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      CHARACTER FILNM*5
C
C==   initialize variables and arrays by BLOCK DATA INIT
C
      CALL INPUT(NDIST,NTOT,VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS,GAMM,
     *   PRESS,FILNM,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,DBANG)
C
      IABORT = 0
      CALL STRSTN(NDIST,NTOT,NPTS,VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *   GAMM,PRESS,DILATO,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,STRNP,
     *   DBANG,IABORT)
C
C==   write out data
      IF(IWRT.GT.0) CALL ENDWRT(NDIST,NTOT,VOLSMP,GAMM,FDBND,YMULT,
     *   IKIND,IMORI,IPOIS,PRESS,IAUG,FILNM,STSFUD,IWRT)
C
```

```
      CALL RESWRT(NDIST,NTOT,VOLSMP,GAMM,FDBND,YMULT,IKIND,
     *   IMORI,IPOIS,PRESS,IAUG,FILNM,STSFUD,IWRT)
C
      END
C
C
      SUBROUTINE INPUT(NDIST,NTOT,VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *   GAMM,PRESS,FILNM,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,DBANG)
C==== reads in problem input either by file or keyboard. if data entered
C     through a file, user inputs name only, a file extension of DAT is
C     assumed. the first line in the input file is used for a user
C     heading and is not read in, constituent material properties
C     designated as follows: 1-inclusion,2-void or vacuole,3-matrix
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 250,NPTMX = 250,NTDIS = 1)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /PROPD/ YMPARM(4),CVPARM(3),FBPARM(3)
      COMMON /VERIF/ VSTRN(NPTMX),VSTRS(NPTMX),VDVV(NPTMX),
     *   VTSTRS(NPTMX),EDIFF(NPTMX),VDIFF(NPTMX),VPOIS(NPTMX),
     *   PDIFF(NPTMX)
      CHARACTER ANS*1,FILNM*5
C
      WRITE (6,'(/,A)') ' Read data from file? (Y/N)'
      READ (5,'(A1)') ANS
C
      IF (ANS.EQ.'Y') THEN
         WRITE (6,'(A)') ' File to read? (.INP will be appended)'
         READ (5,'(A5)') FILNM
         OPEN (UNIT=7,FILE=FILNM//'.INP',FORM='FORMATTED',STATUS='OLD')
```

```
      READ (7,*)
      READ (7,*) NTOT
      READ (7,*) NDIST
      DO 10 I = 1,ABS(NDIST)
         READ (7,*) RADAVG(I)
         READ (7,*) LOGSTD(I)
         READ (7,*) VLFRFO(I),VLFRVO(I)
10    CONTINUE
      READ (7,*) VOLSMP
      READ (7,*) FDBND,YMULT,IKIND,IMORI,IPOIS
      READ (7,*) G(3),G(1)
      READ (7,*) POIS(3),POIS(1)
      READ (7,*) G(2),K(2)
      READ (7,*) GAMM,PRESS
      READ (7,*) IAUG,STNTOL,ITERMX
      READ (7,*) NLTOT,STSFUD,DBANG,ISKIP
      READ (7,*) AKO(1),AK1(1),AK2(1)
      READ (7,*) YMPARM(1),YMPARM(2),YMPARM(3),YMPARM(4)
      READ (7,*) CVPARM(1),CVPARM(2),CVPARM(3)
      READ (7,*) FBPARM(1),FBPARM(2),FBPARM(3)
      CLOSE (7)
   ELSE
      WRITE (6,'(/,A,I3,A)')
     *    ' no. pts desired in stress-strain curve (<',GSMX,')'
      READ (5,*) NTOT
      WRITE (6,'(A,I1,A)') ' no. of particle distributions (<=',
     *    NTDIS,')'
      READ (5,*) NDIST
      DO 20 I = 1,ABS(NDIST)
         WRITE (6,'(A,I1,A)') ' for distribution no. ',I,
     *       ' mean radius (micron)'
         READ (5,*) RADAVG(I)
         WRITE (6,'(A)') ' log normal radius std dev'
         READ (5,*) LOGSTD(I)
         WRITE (6,'(A)') ' initial volume fraction filler and void'
         READ (5,*) VLFRFO(I),VLFRVO(I)
20    CONTINUE
```

```
      WRITE (6,'(A)') ' sample volume (mm3)'
      READ (5,*) VOLSMP
      WRITE (6,'(A)') ' dbnd frac,rad dist mult,w-type,m-type,v-type'
      READ (5,*) FDBND,YMULT,IKIND,IMORI,IPOIS
      WRITE (6,'(A)') ' matrix and filler shear modulus (Pa)'
      READ (5,*) G(3),G(1)
      WRITE (6,'(A)') ' matrix and filler Poisson ratio'
      READ (5,*) POIS(3),POIS(1)
      WRITE (6,'(A)') ' void shear and bulk modulus (Pa)'
      READ (5,*) G(2),K(2)
      WRITE (6,'(A)') ' Gc (J/m2) and applied pressure (Pa)'
      READ (5,*) GAMM,PRESS
      WRITE (6,'(A)') ' a-type, strain tolerance, max. iterations'
      READ (5,*) IAUG,STNTOL,ITERMX
      WRITE (6,'(A)') ' # pts before debnd, crit strs(Pa), debond ang
     *(deg), avg.frc.'
      READ (5,*) NLTOT,STSFUD,DBANG,ISKIP
      WRITE (6,'(A)') ' matrix tensile modulus coeffs k0,k1,k2 (Pa)'
      READ (5,*) AK0(1),AK1(1),AK2(1)
      WRITE (6,'(A)') ' Ym step, Ym max iter, Ym %err, start index'
      READ (5,*) YMPARM(1),YMPARM(2),YMPARM(3),YMPARM(4)
      WRITE (6,'(A)') ' Cv step size, Cv %err, start index'
      READ (5,*) CVPARM(1),CVPARM(2),CVPARM(3)
      WRITE (6,'(A)') ' Fb step, Fb max iter, Fb %err'
      READ (5,*) FBPARM(1),FBPARM(2),FBPARM(3)
      FILNM = 'DEFAULT'
    ENDIF
C
C==   set write file flag, 0=STRWRT, 1=STRWRT,DBGWRT,HSTWRT, 2=all
      IWRT = 0
      IF (NDIST.LT.0) IWRT = 1
      NDIST = ABS(NDIST)
      IF (IAUG.EQ.0) NLTOT = 0
C
C     read datafile containing stress-strain verification data
C
      NTOT = 0
```

```
      WRITE (6,'(A,A8)') ' Reading Data File: ',FILNM
      OPEN (UNIT=7,FILE=FILNM//'.DAT',FORM='FORMATTED',STATUS='OLD')
      READ (7,*)
      DO 30 I = 1,NPTMX
         READ (7,*,END=40) VSTRN(I),VSTRS(I),VDVV(I),VTSTRS(I)
         VSTRS(I) = VSTRS(I)*1E6
         VTSTRS(I) = VTSTRS(I)*1E6
         NTOT = NTOT+1
   30 CONTINUE
   40 CONTINUE
      CLOSE (7)
C
      RETURN
      END
C
C
      SUBROUTINE STRSTN(NDIST,NTOT,NPTS,VOLSMP,FDBND,YMULT,IKIND,IMORI,
     *   IPOIS,GAMM,PRESS,DILATO,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,
     *   STRNP,DBANG,IABORT)
C==== main subroutine which organizes finding optimal Ym
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 250,NPTMX = 250,NTDIS = 1)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /VERIF/ VSTRN(NPTMX),VSTRS(NPTMX),VDVV(NPTMX),
     *   VTSTRS(NPTMX),EDIFF(NPTMX),VDIFF(NPTMX),VPOIS(NPTMX),
     *   PDIFF(NPTMX)
C
C     initialize abort flag
      IABORT = 0
C     find index that separates no debonding and debonding phases
      DO 10 I = 1,NTOT
```

```
         IF (VTSTRS(I).LE.STSFUD) ICUTOF = I
   10 CONTINUE
C
      CALL YMFIND(ICUTOF,FDBND,YMULT,IKIND,IMORI,IPOIS,IAUG,STNTOL,
     *   ITERMX,IABORT)
C
C     CALL CVPLT(ICUTOF,IABORT)
C
      CALL CVFIND(NTOT,ICUTOF,FDBND,YMULT,IKIND,IMORI,IPOIS,IAUG,STNTOL,
     *   ITERMX,IABORT)
C
C     CALL CVPLT(NTOT,IABORT)
C
      NDIST = ICUTOF
C
      RETURN
      END
C
C
      SUBROUTINE YMFIND(ICUTOF,FDBND,YMULT,IKIND,IMORI,IPOIS,IAUG,
     *   STNTOL,ITERMX,IABORT)
C==== subroutine finds the particle interaction factor that will give
C     lowest overall error when comparing theoretical E to expt E.
C     convert dilatation data to Poisson data to make comparison.
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 250,NPTMX = 250,NTDIS = 1)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
```

```
      COMMON /PROPD/ YMPARM(4),CVPARM(3),FBPARM(3)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      COMMON /VERIF/ VSTRN(NPTMX),VSTRS(NPTMX),VDVV(NPTMX),
     *   VTSTRS(NPTMX),EDIFF(NPTMX),VDIFF(NPTMX),VPOIS(NPTMX),
     *   PDIFF(NPTMX)
C
      IF (IABORT.EQ.1) RETURN
C
      YMSTEP = YMPARM(1)
      IUPDWN = 0
      IHOLD = 0
      ICNT = 1
      ISTART = INT(YMPARM(4))
      CONCI = VLFRFO(1)
      CONCV = VLFRVO(1)
      STRNO = VSTRN(ISTART)
      ECMPX = VTSTRS(ISTART)/VSTRN(ISTART)
      CALL MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,IPOIS,STRNO,
     *   IAUG,IABORT)
      STRNPO = ECMPX/E(3)*STRNO
C
      WRITE (6,'(/,A,I3)') ' Number of points in initial phase=',ICUTOF
      WRITE (6,5000)
      ICVEND = INT(YMPARM(2))
      DO 50 ICVRG = 1,ICVEND
C
         DO 30 ICNT = ISTART,ICUTOF
            STRNO = VSTRN(ICNT)
            ECMPX = VTSTRS(ICNT+1)/VSTRN(ICNT+1)
            DO 10 JTER = 1,ITERMX
               CALL MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,
     *            IPOIS,STRNPO,IAUG,IABORT)
               STRNP1(ICNT) = (ECMP(ICNT)/E(3))*STRNO
               ERR = 1
               IF (STRNP1(ICNT).NE.0) ERR =
```

```
     *            ABS((STRNP1(ICNT)-STRNP0)/STRNP1(ICNT))
                  IF (ERR.LE.STNTOL) GOTO 20
                  STRNP0 = STRNP1(ICNT)
   10          CONTINUE
               WRITE (6,'(A)')
     *            ' SBR YMFIND: matrix strain iteration max. reached.'
               IABORT = 1
               IF (IABORT.EQ.1) RETURN
   20          CONTINUE
               EDIFF(ICNT) = (ECMP(ICNT)/ECMPX)-1
               NETVF(ICNT) = CONCI
               NETVV(ICNT) = CONCV
   30       CONTINUE
C
C     use array PRBSRV to temporarily hold EDIFF data for calc of stats
            IDUM = 0
            DO 40 I = ISTART,ICUTOF
               IDUM = IDUM+1
               PRBSRV(IDUM) = EDIFF(I)
   40       CONTINUE
C
            EMAX = FMAX(PRBSRV,IDUM)
            EMIN = FMIN(PRBSRV,IDUM)
            CALL MOMENT(PRBSRV,IDUM,AVE,ADEV,SDEV,VAR,SKEW,CURT)
            WRITE (6,5100) YMULT,AVE,SDEV,VAR,EMAX,EMIN
C
            IF (ABS(AVE).LE.YMPARM(3)) GOTO 60
            IF (AVE.GE.0.AND.ICVEND.GT.1) THEN
               IUPDWN = 0
               IF (IUPDWN.NE.IHOLD) YMSTEP = 0.1*YMSTEP
               YMULT = YMULT-YMSTEP
            ELSEIF (AVE.LT.0.AND.ICVEND.GT.1) THEN
               IUPDWN = 1
               IF (IUPDWN.NE.IHOLD) YMSTEP = 0.1*YMSTEP
               YMULT = YMULT+YMSTEP
            ELSE
            ENDIF
```

```
         IHOLD = IUPDWN
C
   50 CONTINUE
      IF(ICVEND.GT.1)THEN
         WRITE (6,'(A)') ' SBR YMFIND: Ym iteration max. reached.'
         IABORT = 1
      ENDIF
   60 CONTINUE
      WRITE (6,'(A,E11.4)') ' SBR YMFIND: optimum YMULT= ',YMULT
C
      DO 70 ICNT = ISTART+1,ICUTOF
         DILAT(ICNT) = (1-2.0*POISC(ICNT-1))*VSTRN(ICNT)
         STRESS(ICNT) = ECMP(ICNT-1)*VSTRN(ICNT)
         VDIFF(ICNT) = DILAT(ICNT)-VDVV(ICNT)
         VPOIS(ICNT) = ((VDVV(ICNT)/VSTRN(ICNT))-1)/(-2.0)
         IF (VPOIS(ICNT).NE.0) PDIFF(ICNT) = (POISC(ICNT)/VPOIS(ICNT))-1
   70 CONTINUE
C
      RETURN
 5000 FORMAT (/,'     YMULT     avg Edif       sdev        var          m
     *ax         min')
 5100 FORMAT (1X,1P6E12.4)
      END
C
C
      SUBROUTINE CVFIND(NTOT,ICUTOF,FDBND,YMULT,IKIND,IMORI,IPOIS,IAUG,
     *   STNTOL,ITERMX,IABORT)
CXXX= subroutine finds the void fraction that will give
C     lowest overall error when comparing theoretical E to expt E.
C     convert dilatation data to Poisson data to make comparison.
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 250,NPTMX = 250,NTDIS = 1)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
```

```
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /PROPD/ YMPARM(4),CVPARM(3),FBPARM(3)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      COMMON /VERIF/ VSTRN(NPTMX),VSTRS(NPTMX),VDVV(NPTMX),
     *   VTSTRS(NPTMX),EDIFF(NPTMX),VDIFF(NPTMX),VPOIS(NPTMX),
     *   PDIFF(NPTMX)
C
      IF (IABORT.EQ.1) RETURN
C
      ISTART = INT(YMPARM(4))
      FBSTEP = FBPARM(1)
      ICVEND = INT(VLFRFO(1)/CVPARM(1))
      IFBEND = INT(FBPARM(2))
      WRITE (6,'(/,A,I3)') ' Number of points in debond phase= ',NTOT-
     *   ICUTOF
      WRITE (6,5000)
C
      DO 70 ICVRG = 1,IFBEND
         ICVBEG = 1
         IUPDWN = 0
         IHOLD = 0
         CONCI = VLFRFO(1)
         CONCV = VLFRVO(1)
C
         DO 50 ICNT = ICUTOF+1,NTOT-1
            STRNO = VSTRN(ICNT)
            ECMPX = VTSTRS(ICNT+1)/VSTRN(ICNT+1)
            CALL MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *         STRNO,IAUG,IABORT)
            STRNPO = ECMPX/E(3)*STRNO
```

```
C
            DO 30 ICV = ICVBEG,ICVEND
               CONCV = CONCV+CVPARM(1)
               CONCI = CONCI-CVPARM(1)
               DO 10 JTER = 1,ITERMX
                  CALL MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,
     *               IPOIS,STRNPO,IAUG,IABORT)
                  STRNP1(ICNT) = (ECMP(ICNT)/E(3))*STRNO
                  ERR = 1
                  IF (STRNP1(ICNT).NE.0) ERR =
     *               ABS((STRNP1(ICNT)-STRNPO)/STRNP1(ICNT))
                  IF (ERR.LE.STNTOL) GOTO 20
                  STRNPO = STRNP1(ICNT)
   10          CONTINUE
               WRITE (6,'(A)')
     *            ' SBR CVFIND: matrix strain iteration max. reached.'
               IABORT = 1
               IF (IABORT.EQ.1) RETURN
   20          CONTINUE
C
               EDIFF(ICNT) = (ECMP(ICNT)/ECMPX)-1
               IF (EDIFF(ICNT).LT.0) THEN
                  WRITE (6,'(A)') ' SBR CVFIND:increase Ym as required.'
                  IABORT = 1
                  RETURN
               ELSEIF (ABS(EDIFF(ICNT)).LE.CVPARM(2)) THEN
                  GOTO 40
               ELSE
               ENDIF
C
   30       CONTINUE
C     overstepped Fb that permits solution, go modify Fb and redo
            IHOLD = 1
            GOTO 90
C
   40       CONTINUE
            ICVBEG = ICV+1
```

```
            NETVV(ICNT) = CONCV
            NETVF(ICNT) = CONCI
            DILAT(ICNT) = (1-2.0*POISC(ICNT-1))*VSTRN(ICNT)
            STRESS(ICNT) = ECMP(ICNT-1)*VSTRN(ICNT)
C           IF (VDVV(ICNT).NE.0) VDIFF(ICNT) = (DILAT(ICNT)/
C     *        VDVV(ICNT))-1
            VDIFF(ICNT) = DILAT(ICNT)-VDVV(ICNT)
            VPOIS(ICNT) = ((VDVV(ICNT)/STRNO)-1)/(-2.0)
            IF (VPOIS(ICNT).NE.0) PDIFF(ICNT) = (POISC(ICNT)/
     *        VPOIS(ICNT))-1
   50     CONTINUE
C
C     use array PRBSRV to temporarily hold VDIFF data for calc of stats
          IDUM = 0
          DO 60 I = INT(ICUTOF+CVPARM(3)),NTOT-1
             IDUM = IDUM+1
             PRBSRV(IDUM) = VDIFF(I)
   60     CONTINUE
C
          VMAX = FMAX(PRBSRV,IDUM)
          VMIN = FMIN(PRBSRV,IDUM)
          CALL MOMENT(PRBSRV,IDUM,AVE,ADEV,SDEV,VAR,SKEW,CURT)
          VFREM = CONCI/VLFRFO(1)
          WRITE (6,5100) FDBND,AVE,VFREM,SDEV,VMAX,VMIN
C
          IF (ABS(AVE).LE.FBPARM(3)) GOTO 80
   90     CONTINUE
          IF (IFBEND.GT.1) THEN
             IF (IUPDWN.NE.IHOLD)THEN
                FDBND = FDBND-FBSTEP
                FBSTEP = 0.1*FBSTEP
             ENDIF
             FDBND = FDBND+FBSTEP
          ENDIF
C
   70 CONTINUE
      IF (IFBEND.GT.1) WRITE (6,'(A)') ' SBR CVFIND: Fb iteration max. r
```

```
     *eached.'
   80 CONTINUE
      WRITE (6,'(A,E11.4)') ' SBR CVFIND: optimum Fb = ',FDBND
C
      RETURN
 5000 FORMAT (/,'       Fb       avg Vdif       fr_Ci        sdev          m
     *ax          min')
 5100 FORMAT (1X,1P6E12.4)
      END
C
C
      SUBROUTINE MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *   STNTMP,IAUG,IABORT)
C==== program for calculating composite modulus based on Mori-Tanaka.
C     FDBND=fraction debond for orthotropic properties in loading
C     direction, IKIND=use inclusion or void properties in calc of
C     Wv matrix, IMORI=type of particle interaction used 0=none,
C     1=inclusion, 2=inclusion and void or vacuole,IPOIS=type of
C     debond properties 0=orthotropic,1=isotropic. recalculates
C     matrix modulus each time routine is called.
C
      REAL IDENT,K,KCMP,MAG
      PARAMETER (GSMX = 250,NPTMX = 250,NTDIS = 1)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      DIMENSION CAVG(3,3)
C
      IF (IABORT.EQ.1) RETURN
C
C     changed this block so that any variables that depend on matrix
C     modulus calculated each time new strain is available
      CALL CALCIO(STNTMP,IAUG,IABORT)
      CALL CALCCV(FDBND,IPOIS,IABORT)
      CALL CMPRPO(IKIND,IMORI,IABORT)
C
      CALL CMPRP(CONCI,CONCV,YMULT,CAVG,IABORT)
      C11(ICNT) = CAVG(1,1)
```

```
      C12(ICNT) = CAVG(1,2)
      C21(ICNT) = CAVG(2,1)
      C22(ICNT) = CAVG(2,2)
      C23(ICNT) = CAVG(2,3)
      ECMP(ICNT) = C11(ICNT)-2.0*C12(ICNT)*C21(ICNT)/(C22(ICNT)+
     *   C23(ICNT))
      POISC(ICNT) = C21(ICNT)/(C22(ICNT)+C23(ICNT))
C
      RETURN
      END
C
C
      BLOCK DATA INIT
C==== initialize all variables and arrays used in program
C     check NPTMX if NTDIS or GSMX are changed.
C        NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 250,NPTMX = 250,NTDIS = 1)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /MATRA/ BETA(2),WI(3,3),WV(3,3),IDENT(3,3)
      COMMON /MATRB/ S(3,3),CA(3,3),CB(3,3),CE(3,3),CF(3,3)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /PROPD/ YMPARM(4),CVPARM(3),FBPARM(3)
      COMMON /VERIF/ VSTRN(NPTMX),VSTRS(NPTMX),VDVV(NPTMX),
     *   VTSTRS(NPTMX),EDIFF(NPTMX),VDIFF(NPTMX),VPOIS(NPTMX),
     *   PDIFF(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
```

```
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
C
      DATA Z /GSMX*0/ RADIUS /NPTMX*0/ PROB /NPTMX*0/
      DATA NUMPAR /NPTMX*0/ VOLPAR /NPTMX*0/ NETVF /NPTMX*0/ NETVV /
     *   NPTMX*0/ DADC /NPTMX*0/ NPARTL /NTDIS*0/
      DATA RADAVG /NTDIS*0/ LOGSTD /NTDIS*0/ VLFRF0 /NTDIS*0/ VLFRV0 /
     *   NTDIS*0/
      DATA BETA /2*0/ WI /9*0/ WV /9*0/ IDENT /1,0,0,0,1,0,0,0,1/
      DATA S /9*0/ CA /9*0/ CB /9*0/ CE /9*0/ CF /9*0/
      DATA K /3*0/ G /3*0/ E /3*0/ POIS /3*0/ CI /9*0/ CV /9*0/ CO /9*0/
      DATA C11 /NPTMX*0/ C12 /NPTMX*0/ C21 /NPTMX*0/ C22 /NPTMX*0/ C23 /
     *   NPTMX*0/ ECMP /NPTMX*0/ POISC /NPTMX*0/
      DATA AK0 /1*0/ AK1 /1*0/ AK2 /1*0/ STRNP1 /NPTMX*0/
      DATA YMPARM /4*0/ CVPARM /3*0/ FBPARM /3*0/
      DATA VSTRN /NPTMX*0/ VSTRS /NPTMX*0/ VDVV /NPTMX*0/ VTSTRS /NPTMX*
     *   0/ EDIFF /NPTMX*0/ VDIFF /NPTMX*0/ VPOIS /NPTMX*0/ PDIFF /NPTMX
     *   *0/
      DATA CRTSTN /NPTMX*0/ STRESS /NPTMX*0/ DILAT /NPTMX*0/ PRBSRV /
     *   NPTMX*0/ SORRAD /NPTMX*0/ SORPAR /NPTMX*0/ SORVLP /NPTMX*0/
     *   IPDIST /NPTMX*0/
C
      END
C
C
      SUBROUTINE CALCIO(STNTMP,IAUG,IABORT)
C==== calculate the property matrix for inclusion and matrix,
C     isotropic relations. have assumed that matrix tensile
C     modulus can be fitted to a 3rd order polynomial.
C
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 250,NPTMX = 250,NTDIS = 1)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPC/ AK0(1),AK1(1),AK2(1),STRNP1(NPTMX)
C
      IF (IABORT.EQ.1) RETURN
```

```
C
C     make matrix shear modulus dependent on strain if NL analysis
C     desired (IAUG>0). chose this form to make compatible with existing
C     program and allow the tensile modulus curve-fitted parameters to
C     be entered.
      IF (IAUG.GT.0) G(3) = (AK0(1)+AK1(1)*STNTMP+AK2(1)*STNTMP**2)/(2.0
     *   *(1+POIS(3)))
C
      K(1) = (2.0*G(1)*(1+POIS(1)))/(3.0*(1.0-2.0*POIS(1)))
      E(1) = G(1)*(2.0*(1+POIS(1)))
      K(3) = (2.0*G(3)*(1+POIS(3)))/(3.0*(1.0-2.0*POIS(3)))
      E(3) = G(3)*(2.0*(1+POIS(3)))
      C1 = K(1)+(4.0/3.0)*G(1)
      C2 = K(1)-(2.0/3.0)*G(1)
      C3 = K(3)+(4.0/3.0)*G(3)
      C4 = K(3)-(2.0/3.0)*G(3)
      DO 20 I = 1,3
         DO 10 J = 1,3
            CI(I,J) = C2
            CO(I,J) = C4
            IF (I.EQ.J) CI(I,J) = C1
            IF (I.EQ.J) CO(I,J) = C3
   10    CONTINUE
   20 CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE CALCCV(FDBND,IPOIS,IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE CMPRPO(IKIND,IMORI,IABORT)
        (subroutine details may be found in Appendix E)
```

```
      RETURN
      END
C
      SUBROUTINE CMPRP(CONCI,CONCV,YMULT,CAVG,IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE CALCW(IKIND,IMORI,IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE CALCS(IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE GAMMA(A,CONC,ITYPE,YMULT,IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE ADD(C,A,B)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE SUB(C,A,B)
        (subroutine details may be found in Appendix E)
      RETURN
      END
```

```
C
C
      SUBROUTINE MULT(C,A,B)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE INVERT(AI,A,IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
C**** function for finding maximum value
      FUNCTION FMAX(DATA,N)
      DIMENSION DATA(N)
      BIG = -1.0E10
C
      DO 10 I = 1,N
         BIG = AMAX1(DATA(I),BIG)
   10 CONTINUE
      FMAX = BIG
C
      RETURN
      END
C
C
      SUBROUTINE ENDWRT(ICUTOF,NTOT,VOLSMP,GAMM,FDBND,YMULT,IKIND,IMORI,
     *   IPOIS,PRESS,IAUG,FILNM,STSFUD,IWRT)
C==== write out strain,E_expt,E_pred,E_diff,Pois_expt,Pois_pred,P_diff
C     TStrs expt,TStrs pred,NetVV
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 250,NPTMX = 250,NTDIS = 1)
```

```
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /PROPD/ YMPARM(4),CVPARM(3),FBPARM(3)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      COMMON /VERIF/ VSTRN(NPTMX),VSTRS(NPTMX),VDVV(NPTMX),
     *   VTSTRS(NPTMX),EDIFF(NPTMX),VDIFF(NPTMX),VPOIS(NPTMX),
     *   PDIFF(NPTMX)
      CHARACTER FILNM*5
C
      IF (FILNM.EQ.'DEFAULT') FILNM = '_STRS'
      WRITE (6,'(/,A,A5,A)') ' Writing to ',FILNM,'O.DAT'
C
      OPEN (UNIT=7,FILE=FILNM//'O.DAT',STATUS='UNKNOWN')
      WRITE (7,5000)
      I = 1
      WRITE (7,'(1X,I1,4(3X,0PE11.4))') I,RADAVG(I),LOGSTD(I),
     *   VLFRFO(I),VLFRVO(I)
C
      WRITE (7,5100) G(3),G(1),POIS(3),POIS(1),G(2),K(2)
      WRITE (7,5200) VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS
      WRITE (7,5300) ICUTOF,GAMM,IAUG,STSFUD
      WRITE (7,5500) AKO(1),AK1(1),AK2(1)
      WRITE (7,5400)
      ISTART = INT(YMPARM(4))
      DO 20 ICNT = ISTART+1,NTOT-1
         ECMPI = VTSTRS(ICNT+1)/VSTRN(ICNT+1)
         WRITE (7,5600) VSTRN(ICNT),ECMPI,ECMP(ICNT),EDIFF(ICNT),
     *      VDVV(ICNT),DILAT(ICNT),VDIFF(ICNT),VPOIS(ICNT),POISC(ICNT),
     *      PDIFF(ICNT),VTSTRS(ICNT),STRESS(ICNT),NETVV(ICNT)
```

```
   20 CONTINUE
C
      CLOSE (7)
      RETURN
 5000 FORMAT (' #    avg Rad(um)    std dev          Vf           Vv')
 5100 FORMAT (' Gm(Pa)=',0PE11.4,' Gf(Pa)=',0PE11.4,' vm=',0PE11.4,
     *   ' vf=',0PE11.4,' Gv(Pa)=',0PE11.4,' Kv=',0PE11.4)
 5200 FORMAT (' V(mm3) =',0PE11.4,' frac dbnd=',0PE11.4,' Y-mult=',0PE11
     *   .4,' w-type=',I3,' m-type=',I3,' v-type=',I3)
 5300 FORMAT (' ICUTOF=',I3,' Gc(Pa-m)=',0PE11.4,' IAUG=',I2,
     *   ' crit.strs(Pa)=',0PE11.4)
 5400 FORMAT ('  crit strn    E_expt(Pa)   E_pred(Pa)    %E_diff      V_
     *expt       V_pred      abs V_diff     v_expt       v_pred       %v
     *_diff     TS_expt      TS_pred       NetVV')
 5500 FORMAT (' AK0(Pa)=',0PE11.4,' AK1(Pa)=',0PE11.4,' AK2(Pa)=',0PE11
     *   .4)
 5600 FORMAT (1X,13(1PE11.4,2X))
C
      END
C
C
      SUBROUTINE RESWRT(ICUTOF,NTOT,VOLSMP,GAMM,FDBND,YMULT,IKIND,IMORI,
     *   IPOIS,PRESS,IAUG,FILNM,STSFUD,IWRT)
C==== write out strain,E_expt,E_pred,E_diff,Pois_expt,Pois_pred,P_diff
C     TStrs expt,TStrs pred,NetVV
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 250,NPTMX = 250,NTDIS = 1)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
```

```
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /PROPD/ YMPARM(4),CVPARM(3),FBPARM(3)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      COMMON /VERIF/ VSTRN(NPTMX),VSTRS(NPTMX),VDVV(NPTMX),
     *   VTSTRS(NPTMX),EDIFF(NPTMX),VDIFF(NPTMX),VPOIS(NPTMX),
     *   PDIFF(NPTMX)
      CHARACTER FILNM*5
C
      IF (FILNM.EQ.'DEFAULT') FILNM = '_STRS'
      WRITE (6,'(/,A,A5,A)') ' Writing to ',FILNM,'R.DAT'
C
      OPEN (UNIT=7,FILE=FILNM//'R.DAT',STATUS='UNKNOWN')
      WRITE (7,5000)
      I = 1
      WRITE (7,'(1X,I1,4(3X,0PE11.4))') I,RADAVG(I),LOGSTD(I),
     *   VLFRFO(I),VLFRVO(I)
C
      WRITE (7,5100) G(3),G(1),POIS(3),POIS(1),G(2),K(2)
      WRITE (7,5200) VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS
      WRITE (7,5300) ICUTOF,GAMM,IAUG,STSFUD
      WRITE (7,5500) AKO(1),AK1(1),AK2(1)
      WRITE (7,5400)
      ISTART = INT(YMPARM(4))
      GCTOT = 0
      IDUM = 0
      DO 20 ICNT = ISTART+1,NTOT-1
         ECMPI = VTSTRS(ICNT+1)/VSTRN(ICNT+1)
         ECMPO = VTSTRS(ICNT)/VSTRN(ICNT)
         DC = NETVF(ICNT)-NETVF(ICNT-1)
         IF(DC.NE.0)THEN
            GCDADC = ((ECMPI-ECMPO)/DC)*VSTRN(ICNT)**2
            GCTOT = GCTOT+GCDADC
            IDUM = IDUM+1
            SORRAD(IDUM) = GCDADC
         ENDIF
```

```
         WRITE (7,5600) VSTRN(ICNT),VTSTRS(ICNT)/1E6,STRESS(ICNT)/1E6,
     *      VDVV(ICNT),DILAT(ICNT),NETVV(ICNT),GCDADC/1E6,ABS(DC),
     *      GCTOT/1E6
   20 CONTINUE
C
      CLOSE (7)
C
      CALL MOMENT(SORRAD,IDUM,AVE,ADEV,SDEV,VAR,SKEW,CURT)
      WRITE(6,'(/,A,1PE11.4,1X,1PE11.4)') ' SBR RESWRT: avg. and tot. en
     *ergy release (MPa):',AVE/1E6,GCTOT/1E6
C
      RETURN
 5000 FORMAT (' #    avg Rad(um)    std dev          Vf           Vv')
 5100 FORMAT (' Gm(Pa)=',0PE11.4,' Gf(Pa)=',0PE11.4,' vm=',0PE11.4,
     *   ' vf=',0PE11.4,' Gv(Pa)=',0PE11.4,' Kv=',0PE11.4)
 5200 FORMAT (' V(mm3) =',0PE11.4,' frac dbnd=',0PE11.4,' Y-mult=',0PE11
     *   .4,' w-type=',I3,' m-type=',I3,' v-type=',I3)
 5300 FORMAT (' ICUTOF=',I3,' Gc(Pa-m)=',0PE11.4,' IAUG=',I2,
     *   ' crit.strs(Pa)=',0PE11.4)
 5400 FORMAT ('  crit strn   TS_exp(MPa)  TS_cal(MPa)     dV_exp       d
     *V_cal       NetVv     GCDADC(MPa)       dC      CumGcAC(MPa)')
 5500 FORMAT (' AK0(Pa)=',0PE11.4,' AK1(Pa)=',0PE11.4,' AK2(Pa)=',0PE11
     *   .4)
 5600 FORMAT (1X,9(1PE11.4,2X))
C
      END
C
C
C**** function for finding minimum value
      FUNCTION FMIN(DATA,N)
      DIMENSION DATA(N)
      SMALL = 1.0E10
C
      DO 10 I = 1,N
         SMALL = AMIN1(DATA(I),SMALL)
   10 CONTINUE
      FMIN = SMALL
```

```
C
      RETURN
      END
C
C
C**** variable identification
C     DATA : array of DATA of length N
C     AVE  : average
C     ADEV : average deviation or mean absolute deviation
C     SDEV : standard deviation
C     VAR  : variance
C     SKEW : skewness
C     CURT : kurtosis
C
      SUBROUTINE MOMENT(DATA,N,AVE,ADEV,SDEV,VAR,SKEW,CURT)
      DIMENSION DATA(N)
      IF (N.LE.1) PAUSE 'N must be at least 2'
      S = 0.
      DO 10 J = 1,N
         S = S+DATA(J)
   10 CONTINUE
      AVE = S/N
      ADEV = 0.
      VAR = 0.
      SKEW = 0.
      CURT = 0.
      DO 20 J = 1,N
         S = DATA(J)-AVE
         ADEV = ADEV+ABS(S)
         P = S*S
         VAR = VAR+P
         P = P*S
         SKEW = SKEW+P
         P = P*S
         CURT = CURT+P
   20 CONTINUE
      ADEV = ADEV/N
```

```
VAR = VAR/(N-1)
SDEV = SQRT(VAR)
IF (VAR.NE.0.) THEN
   SKEW = SKEW/(N*SDEV**3)
   CURT = CURT/(N*VAR**2)-3.
ELSE
   PAUSE 'no skew or kurtosis when zero variance'
ENDIF
RETURN
END
```

FORTRAN Listing for P14-DIST.FOR

```
C==== main program
C      P14-DIST.FOR
C====
C     calculates energy release from experimental results and from
C     assumed debonding of log-normal particle distribution. Compares
C     equivalent energy release from each to find modification function
C     that needs to be applied to the log-normal surface area
C     distribution to force it to conform to the measured results.
C
C     user enters the following information:
C        number of points desired in overall particle distribution
C        number of particle distributions
C        avg radius and std dev of each distribution
C        volume fraction of filler and voids of each distribution
C        sample volume
C        fraction debond, YMULT, w-type, m-type, v-type
C        matrix and filler shear modulus
C        matrix and filler poisson ratio
C        void shear and bulk moduli (if both values zero model as voids,
C          if non-zero, use filler or pseudo-filler values)
C        adhesion energy and applied pressure
C        a-type, convergence tolerance, max. iterations
C        # pts before debonding, critical stress, debond angle, iskip
C        coefficients k0, k1, k2 for fitted matrix modulus
C        Ym step,Ym max iter,Ym %err,start index
C        Cv step,Cv %err,start index
C        Fb step,Fb max iter,Fb abs.err
C
C     %err really means fractional error and not percentage error.
C
C     information may be entered using keyboard or by input data file.
C     two options for printing out intermediate results are available:
C        if values for no pts desired in overall part.dist. and number
C           of particle distributions are negative, data files GAUSS,
C           HISTO and DEBUG are written.
```

```
C        if value for no. of particle distributions is negative and
C           value for overall part.dist. is positive, data file DEBUG
C           is written.
C     if input data was entered using data file, the data file STRESS
C     will be renamed to the input data file's name.
C
C     calculates particle size histogram with corresponding filler
C     volume fraction. uses Z-decrements for particle size
C     determination.
C
C     based correction ratio VRATI=M_G on Cv instead of GCDADC_e because
C     in P14 we need to have the proper amount of energy release to
C     obtain the experimental data.
C
C     last revision: 28 OCT 1996  15H30
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /MATRA/ BETA(2),WI(3,3),WV(3,3),IDENT(3,3)
      COMMON /MATRB/ S(3,3),CA(3,3),CB(3,3),CE(3,3),CF(3,3)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      COMMON /VERIF/ VSTRN(NPTMX),VTSTRS(NPTMX),VDVV(NPTMX),
```

```
     *    VNETVV(NPTMX),VGCTOT(NPTMX),CMBDAT(NPTMX),VRATI(NPTMX),
     *    GCTOT(NPTMX)
      CHARACTER FILNM*8
C
C==   initialize variables and arrays by BLOCK DATA INIT
C
      CALL INPUT(NDIST,NTOT,VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS,GAMM,
     *    PRESS,FILNM,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,DBANG)
C
      IABORT = 0
      CALL STRSTN(NDIST,NTOT,NPTS,VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *    GAMM,PRESS,DILATO,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,
     *    STRNP,DBANG,IABORT)
C
C==   write out debond only results and debug data
      IF (IWRT.GE.1) CALL DBGWRT(NDIST,NPTS,NLTOT,IABORT)
      CALL DBGRAT(NDIST,NPTS,NLTOT,VOLSMP,GAMM,FILNM,FDBND,
     *    YMULT,IKIND,IMORI,IPOIS,STSFUD,NEXMAX,NCLMAX,IABORT)
C
C==   calculate surface area modification function to apply to P14 data
      CALL EURCAL(NDIST,NTOT,NPTS,VOLSMP,FDBND,YMULT,IKIND,IMORI,
     *    IPOIS,GAMM,PRESS,DILATO,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,
     *    STRNP,FILNM,NEXMAX,NCLMAX,IABORT)
      CALL EURWRT(NDIST,NPTS,NLTOT,VOLSMP,GAMM,FILNM,FDBND,YMULT,IKIND,
     *    IMORI,IPOIS,STSFUD,NEXMAX,IABORT)
C
      END
C
C
      SUBROUTINE INPUT(NDIST,NTOT,VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *    GAMM,PRESS,FILNM,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,DBANG)
C==== reads in problem input either by file or keyboard. if data entered
C     through a file, user inputs name only, a file extension of DAT is
C     assumed. the first line in the input file is used for a user
C     heading and is not read in, constituent material properties
C     designated as follows: 1-inclusion,2-void or vacuole,3-matrix
C
```

```
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      CHARACTER ANS*1,FILNM*8
C
      WRITE (6,'(/,A)') ' Read data from file? (Y/N)'
      READ (5,'(A1)') ANS
C
      IF (ANS.EQ.'Y') THEN
         WRITE (6,'(A)') ' File to read? (.INP will be appended)'
         READ (5,'(A8)') FILNM
         OPEN (UNIT=7,FILE=FILNM//'.INP',FORM='FORMATTED',STATUS='OLD')
         READ (7,*)
         READ (7,*) NTOT
         READ (7,*) NDIST
         DO 10 I = 1,ABS(NDIST)
            READ (7,*) RADAVG(I)
            READ (7,*) LOGSTD(I)
            READ (7,*) VLFRFO(I),VLFRVO(I)
   10    CONTINUE
         READ (7,*) VOLSMP
         READ (7,*) FDBND,YMULT,IKIND,IMORI,IPOIS
         READ (7,*) G(3),G(1)
         READ (7,*) POIS(3),POIS(1)
         READ (7,*) G(2),K(2)
         READ (7,*) GAMM,PRESS
         READ (7,*) IAUG,STNTOL,ITERMX
         READ (7,*) NLTOT,STSFUD,DBANG,ISKIP
         READ (7,*) AKO(1),AK1(1),AK2(1)
         CLOSE (7)
```

```
      ELSE
         WRITE (6,'(/,A,I3,A)')
     *      ' no. pts desired in stress-strain curve (<',GSMX,')'
         READ (5,*) NTOT
         WRITE (6,'(A,I1,A)') ' no. of particle distributions (<=',
     *      NTDIS,')'
         READ (5,*) NDIST
         DO 20 I = 1,ABS(NDIST)
            WRITE (6,'(A,I1,A)') ' for distribution no. ',I,
     *         ' mean radius (micron)'
            READ (5,*) RADAVG(I)
            WRITE (6,'(A)') ' log normal radius std dev'
            READ (5,*) LOGSTD(I)
            WRITE (6,'(A)') ' initial volume fraction filler and void'
            READ (5,*) VLFRFO(I),VLFRVO(I)
   20    CONTINUE
         WRITE (6,'(A)') ' sample volume (mm3)'
         READ (5,*) VOLSMP
         WRITE (6,'(A)') ' dbnd frac,rad dist mult,w-type,m-type,v-type'
         READ (5,*) FDBND,YMULT,IKIND,IMORI,IPOIS
         WRITE (6,'(A)') ' matrix and filler shear modulus (Pa)'
         READ (5,*) G(3),G(1)
         WRITE (6,'(A)') ' matrix and filler Poisson ratio'
         READ (5,*) POIS(3),POIS(1)
         WRITE (6,'(A)') ' void shear and bulk modulus (Pa)'
         READ (5,*) G(2),K(2)
         WRITE (6,'(A)') ' Gc (J/m2) and applied pressure (Pa)'
         READ (5,*) GAMM,PRESS
         WRITE (6,'(A)') ' a-type, strain tolerance, max. iterations'
         READ (5,*) IAUG,STNTOL,ITERMX
         WRITE (6,'(A)') ' #pts before debond, crit strs(Pa), debond ang
     *(deg), avg.frc.'
         READ (5,*) NLTOT,STSFUD,DBANG,ISKIP
         WRITE (6,'(A)') ' matrix tensile modulus coeffs k0,k1,k2 (Pa)'
         READ (5,*) AKO(1),AK1(1),AK2(1)
         FILNM = 'DEFAULT'
      ENDIF
```

```
C
C==   set write file flag, 0=STRWRT, 1=STRWRT,DBGWRT,HSTWRT, 2=all
      IWRT = 0
      IF (NTOT.LT.0.AND.NDIST.LT.0) IWRT = 2
      IF (NTOT.GT.0.AND.NDIST.LT.0) IWRT = 1
      NDIST = ABS(NDIST)
      NTOT = ABS(NTOT)
      IF (IAUG.EQ.0) NLTOT=0
C
      RETURN
      END
C
C
      SUBROUTINE STRSTN(NDIST,NTOT,NPTS,VOLSMP,FDBND,YMULT,IKIND,IMORI,
     *   IPOIS,GAMM,PRESS,DILATO,IWRT,IAUG,STNTOL,ITERMX,NLTOT,
     *   STSFUD,STRNP,DBANG,IABORT)
C==== main subroutine which organizes particle size distribution,
C     composite property, critical strain and stress and dilation
C     calculation modules. assume full debonding for calc. of derived
C     Gc.
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
C
C==   initialize abort flag
      IABORT = 0
```

```
C
C==   create gaussian distribution of particle size
      WRITE (6,'(/,A)') ' Generating particle distribution'
      CALL GAUSS(NDIST,NTOT,NPTS,IABORT)
C==   write out gaussian cumulative data
      IF (ABS(IWRT).GE.2) CALL GAUWRT(NDIST,NPTS,IABORT)
C
C==   find size and number of particles to debond
      WRITE (6,'(/,A)') ' Finding particle size and number'
      CALL PARTSZ(NDIST,NPTS,VOLSMP,IABORT)
C==   write out particle size and number histogram
      IF (ABS(IWRT).GE.2) CALL HSTWRT(NDIST,NPTS,IABORT)
C
      WRITE (6,'(/,A)') ' Sorting particle distributions'
      CALL SORTER(NDIST,NPTS,IABORT)
      WRITE (6,'(A)') ' Calculating vol fractions and dA/dc'
      CALL VOLFRC(NDIST,NPTS,VOLSMP,DBANG,IABORT)
C
      RETURN
      END
C
C
      SUBROUTINE DBGRAT(NDIST,NPTS,NLTOT,VOLSMP,GAMM,FILNM,FDBND,
     *   YMULT,IKIND,IMORI,IPOIS,STSFUD,NEXMAX,NCLMAX,IABORT)
C==== write out additional data for debugging purposes
C     must divide DADC by 1E-3 to change units from mm to m
C     radius,no.particles,prbsrv referenced to final state after
C     debonding has taken place. ie. debonded r_i to get to NetVv
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,KTMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
```

```
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPC/ AK0(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      COMMON /VERIF/ VSTRN(NPTMX),VTSTRS(NPTMX),VDVV(NPTMX),
     *   VNETVV(NPTMX),VGCTOT(NPTMX),CMBDAT(NPTMX),VRATI(NPTMX),
     *   GCTOT(NPTMX)
      CHARACTER FILNM*5
C
      IF (IABORT.EQ.1) RETURN
C
C     read P15 R-data, use DILAT array to temporarily hold dCv data
      IDUM = 1
      OPEN (UNIT=7,FILE=FILNM//'R.DAT',STATUS='UNKNOWN')
      READ (7,*)
      READ (7,*)
      READ (7,*)
      READ (7,*)
      READ (7,*)
      READ (7,*)
      READ (7,*)
      DO 10 I = 1,NPTMX
         READ (7,*,END=20) VSTRN(IDUM),VTSTRS(IDUM),A,
     *      VDVV(IDUM),A,VNETVV(IDUM),A,DILAT(IDUM),VGCTOT(IDUM)
         IF (VNETVV(IDUM).GT.0) IDUM = IDUM+1
   10 CONTINUE
   20 CONTINUE
      NEXMAX = IDUM-1
      CLOSE (7)
C
      DMAX = FMAX(DILAT,NEXMAX)
      DMIN = FMIN(DILAT,NEXMAX)
      CALL MOMENT(DILAT,NEXMAX,AVE,ADEV,SDEV,VAR,SKEW,CURT)
```

```
      WRITE(6,'(/,A,I3,A,A5,A)') ' Rows:',NEXMAX,' Stats for ',FILNM,
     *   'R.DAT'
      WRITE(6,5700)
      WRITE (6,5800) AVE,SDEV,VAR,DMAX,DMIN
C
      WRITE (6,'(/,A,A5,A)') ' Writing to ',FILNM,'C.DAT'
C
      OPEN (UNIT=7,FILE=FILNM//'C.DAT',STATUS='UNKNOWN')
      WRITE (7,5000)
      DO 40 I = 1,NDIST
         WRITE (7,'(1X,I1,4(3X,0PE11.4))') I,RADAVG(I),LOGSTD(I),
     *      VLFRFO(I),VLFRVO(I)
   40 CONTINUE
C
      NCLMAX = 0
      WRITE (7,5100) G(3),G(1),POIS(3),POIS(1),G(2),K(2)
      WRITE (7,5200) VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS
      WRITE (7,5300) PRESS,GAMM,DILATO,STSFUD
      WRITE(7,5600) AKO(1),AK1(1),AK2(1)
      WRITE (7,5400)
      IDIST = 1
      GCTOT(1) = 0
      DO 50 IHST = 2,NDIST*NPTS+1
         IF(NETVV(IHST).GT.VNETVV(NEXMAX)) GOTO 30
         NCLMAX = NCLMAX+1
         DNETF = ABS(NETVF(IHST)-NETVF(IHST-1))
         DILAT(IHST-1) = DNETF
         GCDADC = -2.0*GAMM*DADC(IHST)/(VOLSMP*1E-3)
         GCTOT(IHST) = GCTOT(IHST-1)+GCDADC/1E6
         WRITE (7,5500) NETVV(IHST),DNETF,GCDADC/1E6,GCTOT(IHST),
     *      SORRAD(IHST-1),SORPAR(IHST-1),PRBSRV(IHST)
   50 CONTINUE
   30 CONTINUE
C
      CLOSE (7)
C
      DMAX = FMAX(DILAT,NCLMAX)
```

```
      DMIN = FMIN(DILAT,NCLMAX)
      CALL MOMENT(DILAT,NCLMAX,AVE,ADEV,SDEV,VAR,SKEW,CURT)
      WRITE(6,'(/,A,I3,A,A5,A)') ' Rows:',NCLMAX,' Stats for ',
     *   FILNM,'C.DAT'
      WRITE(6,5700)
      WRITE (6,5800) AVE,SDEV,VAR,DMAX,DMIN
C
      WRITE(6,'(/,A,1PE11.4,A,E9.2,A)') ' Derived Gc_calc =',
     *   VGCTOT(NEXMAX)/GCTOT(IHST-1),' at',(NETVF(1)-VNETVV(NEXMAX))/
     *   NETVF(1),' frac.rem.Ci'
C
      RETURN
 5000 FORMAT (' #    avg Rad(um)    std dev          Vf            Vv')
 5100 FORMAT (' Gm(Pa)=',0PE11.4,' Gf(Pa)=',0PE11.4,' vm=',0PE11.4,
     *   ' vf=',0PE11.4,' Gv(Pa)=',0PE11.4,' Kv=',0PE11.4)
 5200 FORMAT (' V(mm3) =',0PE11.4,' frac dbnd=',0PE11.4,' Y-mult=',0PE11
     *   .4,' w-type=',I3,' m-type=',I3,' v-type=',I3)
 5300 FORMAT (' P0(Pa)=',0PE11.4,' Gc(Pa-m)=',0PE11.4,' (dV/V)0=',0PE11
     *   .4, ' crit.strs(Pa)=',0PE11.4)
 5400 FORMAT ('    NetVv          dC      GCDADC(MPa)  CumGCDA(MPa)   ra
     *d(mm)      No.Part.     Pr|surv')
 5500 FORMAT (1X,7(1PE11.4,2X))
 5600 FORMAT (' AK0(Pa)=',0PE11.4,' AK1(Pa)=',0PE11.4,' AK2(Pa)=',
     *   0PE11.4)
 5700 FORMAT ('    avg dCv       sdev        var         max         min
     *')
 5800 FORMAT (1X,1P5E12.4)
      END
C
C
      SUBROUTINE EURCAL(NDIST,NTOT,NPTS,VOLSMP,FDBND,YMULT,IKIND,IMORI,
     *   IPOIS,GAMM,PRESS,DILATO,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,
     *   STRNP,FILNM,NEXMAX,NCLMAX,IABORT)
C==== calculates the modification function that needs to be applied to
C     the P14 log-normal particle surface area distribution to get the
C     experimentally determined energy release. the function is based on
C     the ratio of SGAD_exp/SGAD_cal.
```

```
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1,PRERR = 0.02)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /VERIF/ VSTRN(NPTMX),VTSTRS(NPTMX),VDVV(NPTMX),
     *   VNETVV(NPTMX),VGCTOT(NPTMX),CMBDAT(NPTMX),VRATI(NPTMX),
     *   GCTOT(NPTMX)
      CHARACTER FILNM*5
C
C**   combine P15 and log-normal distribution data based on VNETVV
C
      JSTRT = 1
      IDUM = 0
      IERCNT = 0
      DO 70 J = 1,NEXMAX
         DO 50 JDUM = JSTRT,NCLMAX
            IF (NETVV(JDUM).GT.VNETVV(J)) THEN
               ADIFF = ABS((NETVV(JDUM)-VNETVV(J)))/VNETVV(J)
               BDIFF = ABS((NETVV(JDUM-1)-VNETVV(J)))/VNETVV(J)
               IF (ADIFF.LT.BDIFF) THEN
                  IDUM = IDUM+1
                  CMBDAT(IDUM) = GCTOT(JDUM)
                  JSTRT = JDUM
                  IF (ADIFF.GT.PRERR) IERCNT = IERCNT+1
                  GOTO 60
               ELSE
                  IDUM = IDUM+1
                  CMBDAT(IDUM) = GCTOT(JDUM-1)
                  JSTRT = JDUM-1
                  IF (BDIFF.GT.PRERR) IERCNT = IERCNT+1
                  GOTO 60
               ENDIF
            ELSE
            ENDIF
```

```
   50    CONTINUE
   60    CONTINUE
   70 CONTINUE
      NEXMAX = IDUM
C
      WRITE (6,'(/,A,E11.4,A,I4)') ' SBR EURCAL: Total iterations over',
     *   PRERR,' is',IERCNT
C
      DO 80 J = 1,NEXMAX
         VRATI(J) = 0
         IF(CMBDAT(J).NE.0) VRATI(J) = VGCTOT(J)/CMBDAT(J)
   80 CONTINUE
C
      DMAX = FMAX(VRATI,NEXMAX)
      DMIN = FMIN(VRATI,NEXMAX)
      CALL MOMENT(VRATI,NEXMAX,AVE,ADEV,SDEV,VAR,SKEW,CURT)
      WRITE (6,5000)
      WRITE (6,5100) AVE,SDEV,VAR,DMAX,DMIN
C
      RETURN
 5000 FORMAT (
     *   '   avg SG_rat     sdev        var         max         min')
 5100 FORMAT (1X,1P5E12.4)
      END
C
C
      SUBROUTINE EURWRT(NDIST,NPTS,NLTOT,VOLSMP,GAMM,FILNM,FDBND,YMULT,
     *   IKIND,IMORI,IPOIS,STSFUD,NEXMAX,IABORT)
C==== write out data
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,KTMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
```

```
     *   VLFRVO(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /VERIF/ VSTRN(NPTMX),VTSTRS(NPTMX),VDVV(NPTMX),
     *   VNETVV(NPTMX),VGCTOT(NPTMX),CMBDAT(NPTMX),VRATI(NPTMX),
     *   GCTOT(NPTMX)
      CHARACTER FILNM*5
C
      IF (IABORT.EQ.1) RETURN
      WRITE (6,'(/,A,A5,A)') ' Writing to ',FILNM,'E.DAT'
C
      OPEN (UNIT=7,FILE=FILNM//'E.DAT',STATUS='UNKNOWN')
      WRITE (7,5000)
      DO 10 I = 1,NDIST
         WRITE (7,'(1X,I1,4(3X,0PE11.4))') I,RADAVG(I),LOGSTD(I),
     *      VLFRFO(I),VLFRVO(I)
   10 CONTINUE
C
      WRITE (7,5100) G(3),G(1),POIS(3),POIS(1),G(2),K(2)
      WRITE (7,5200) VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS
      WRITE (7,5300) PRESS,GAMM,DILATO,STSFUD
      WRITE (7,5600) AKO(1),AK1(1),AK2(1)
      WRITE (7,5400)
      DO 20 I = 1,NEXMAX
         WRITE (7,5500) VSTRN(I),VNETVV(I),VGCTOT(I),CMBDAT(I),
     *      VRATI(I),VTSTRS(I),VDVV(I)
   20 CONTINUE
C
      CLOSE (7)
C
      RETURN
 5000 FORMAT (' #    avg Rad(um)    std dev          Vf           Vv')
 5100 FORMAT (' Gm(Pa)=',0PE11.4,' Gf(Pa)=',0PE11.4,' vm=',0PE11.4,
     *   ' vf=',0PE11.4,' Gv(Pa)=',0PE11.4,' Kv=',0PE11.4)
 5200 FORMAT (' V(mm3) =',0PE11.4,' frac dbnd=',0PE11.4,' Y-mult=',0PE11
     *   .4,' w-type=',I3,' m-type=',I3,' v-type=',I3)
 5300 FORMAT (' PO(Pa)=',0PE11.4,' Gc(Pa-m)=',0PE11.4,' (dV/V)0=',0PE11
```

```
     *   .4,' crit.strs(Pa)=',0PE11.4)
 5400 FORMAT ('    strn_e       Cv_exp    SGAD_e(MPa)  SGAD_c(MPa)    SG
     *Ratio     TStse(MPa)      dV/V_e')
 5500 FORMAT (1X,7(1PE11.4,2X))
 5600 FORMAT (' AK0(Pa)=',0PE11.4,' AK1(Pa)=',0PE11.4,' AK2(Pa)=',0PE11
     *   .4)
      END
C
C
      SUBROUTINE GAUSS(NDIST,NTOT,NPTS,IABORT)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
      FUNCTION FUNC(X)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE PARTSZ(NDIST,NPTS,VOLSMP,IABORT)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE SORTER(NDIST,NPTS,IABORT)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE VOLFRC(NDIST,NPTS,VOLSMP,DBANG,IABORT)
C==== calculates dA/dc, net Vf, net Vv and probility of survival for
C     given particle radius. Note: net Vf is based on total sample vol.
C     Prob of surv is based on numbers of particles.
C
```

```
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      PARAMETER (PI = 3.1415927)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
C
      IF (IABORT.EQ.1) RETURN
C
C==   calculate total volume fraction filler and void
      NETVF(1) = 0
      NETVV(1) = 0
      DO 10 I = 1,NDIST
         NETVF(1) = NETVF(1)+VLFRFO(I)
         NETVV(1) = NETVV(1)+VLFRVO(I)
   10 CONTINUE
      PRBSRV(1) = 1.0
C
C==   calculate net Vf and Vv, dA/dc and Prob surv.
C
      SRVNUM = 0
C
C==   find total number of particles
      DO 20 ICNT = 1,NDIST
         TLNUMP = TLNUMP+NPARTL(ICNT)
   20 CONTINUE
C
      DO 30 JCNT = 2,NDIST*NPTS+1
         NETVF(JCNT) = NETVF(JCNT-1)-SORVLP(JCNT-1)/VOLSMP
         NETVV(JCNT) = NETVV(JCNT-1)+SORVLP(JCNT-1)/VOLSMP
```

```
         SRVNUM = SRVNUM+SORPAR(JCNT-1)
         PRBSRV(JCNT) = (TLNUMP-SRVNUM)/TLNUMP
         DADC(JCNT) = -6.0*(1-SIN(DBANG*3.14159/180))*VOLSMP/
     *      SORRAD(JCNT-1)
   30 CONTINUE
C
      RETURN
      END
C
C
      BLOCK DATA INIT
C==== initialize all variables and arrays used in program
C     check NPTMX if NTDIS or GSMX are changed.
C        NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /MATRA/ BETA(2),WI(3,3),WV(3,3),IDENT(3,3)
      COMMON /MATRB/ S(3,3),CA(3,3),CB(3,3),CE(3,3),CF(3,3)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      COMMON /VERIF/ VSTRN(NPTMX),VTSTRS(NPTMX),VDVV(NPTMX),
     *   VNETVV(NPTMX),VGCTOT(NPTMX),CMBDAT(NPTMX),VRATI(NPTMX),
     *   GCTOT(NPTMX)
C
```

```
      DATA Z /GSMX*0/ RADIUS /NPTMX*0/ PROB /NPTMX*0/
      DATA NUMPAR /NPTMX*0/ VOLPAR /NPTMX*0/ NETVF /NPTMX*0/ NETVV /
     *   NPTMX*0/ DADC /NPTMX*0/ NPARTL /NTDIS*0/
      DATA RADAVG /NTDIS*0/ LOGSTD /NTDIS*0/ VLFRF0 /NTDIS*0/ VLFRV0 /
     *   NTDIS*0/
      DATA BETA /2*0/ WI /9*0/ WV /9*0/ IDENT /1,0,0,0,1,0,0,0,1/
      DATA S /9*0/ CA /9*0/ CB /9*0/ CE /9*0/ CF /9*0/
      DATA K /3*0/ G /3*0/ E /3*0/ POIS /3*0/ CI /9*0/ CV /9*0/ CO /9*0/
      DATA C11 /NPTMX*0/ C12 /NPTMX*0/ C21 /NPTMX*0/ C22 /NPTMX*0/ C23 /
     *   NPTMX*0/ ECMP /NPTMX*0/ POISC /NPTMX*0/
      DATA AK0 /1*0/ AK1 /1*0/ AK2 /1*0/ STRNP1 /NPTMX*0/
      DATA CRTSTN /NPTMX*0/ STRESS /NPTMX*0/ DILAT /NPTMX*0/ PRBSRV /
     *   NPTMX*0/ SORRAD /NPTMX*0/ SORPAR /NPTMX*0/ SORVLP /NPTMX*0/
     *   IPDIST /NPTMX*0/
      DATA VSTRN /NPTMX*0/ VTSTRS /NPTMX*0/ VDVV /NPTMX*0/
     *   VNETVV /NPTMX*0/ VGCTOT /NPTMX*0/ CMBDAT /NPTMX*0/
     *   VRATI /NPTMX*0/ GCTOT /NPTMX*0/
C
      END
C
C
      SUBROUTINE QSIMP(FUNC,A,B,S)
       (subroutine details may be found in Appendix E)
      END
C
C
      SUBROUTINE TRAPZD(FUNC,A,B,S,N)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE SORT3(N,RA,RB,RC,IRD)
       (subroutine details may be found in Appendix E)
      END
C
C
```

```
      SUBROUTINE GAUWRT(NDIST,NPTS,IABORT)
       (subroutine details may be found in Appendix E)
      END
C
C
      SUBROUTINE HSTWRT(NDIST,NPTS,IABORT)
       (subroutine details may be found in Appendix E)
      END
C
C
      SUBROUTINE DBGWRT(NDIST,NPTS,NLTOT,IABORT)
       (subroutine details may be found in Appendix E)
      END
C
C
C**** function for finding maximum value
      FUNCTION FMAX(DATA,N)
      DIMENSION DATA(N)
      BIG = -1.0E10
C
      DO 10 I = 1,N
         BIG = AMAX1(DATA(I),BIG)
   10 CONTINUE
      FMAX = BIG
C
      RETURN
      END
C
C
C**** function for finding minimum value
      FUNCTION FMIN(DATA,N)
      DIMENSION DATA(N)
      SMALL = 1.0E10
C
      DO 10 I = 1,N
         SMALL = AMIN1(DATA(I),SMALL)
   10 CONTINUE
```

```
      FMIN = SMALL
C
      RETURN
      END
C
C
C**** variable identification
C     DATA : array of DATA of length N
C     AVE  : average
C     ADEV : average deviation or mean absolute deviation
C     SDEV : standard deviation
C     VAR  : variance
C     SKEW : skewness
C     CURT : kurtosis
C
      SUBROUTINE MOMENT(DATA,N,AVE,ADEV,SDEV,VAR,SKEW,CURT)
      DIMENSION DATA(N)
      IF (N.LE.1) PAUSE 'N must be at least 2'
      S = 0.
      DO 10 J = 1,N
         S = S+DATA(J)
   10 CONTINUE
      AVE = S/N
      ADEV = 0.
      VAR = 0.
      SKEW = 0.
      CURT = 0.
      DO 20 J = 1,N
         S = DATA(J)-AVE
         ADEV = ADEV+ABS(S)
         P = S*S
         VAR = VAR+P
         P = P*S
         SKEW = SKEW+P
         P = P*S
         CURT = CURT+P
   20 CONTINUE
```

```
ADEV = ADEV/N
VAR = VAR/(N-1)
SDEV = SQRT(VAR)
IF (VAR.NE.0.) THEN
   SKEW = SKEW/(N*SDEV**3)
   CURT = CURT/(N*VAR**2)-3.
ELSE
   PAUSE 'no skew or kurtosis when zero variance'
ENDIF
RETURN
END
```

APPENDIX C

RESULTS OF GLASS BEAD/HTPB ANALYSES

The following graphs show the predictions of true stress-strain and dilatation-strain behavior using the back-calculated model parameters. Graphs of cumulative strain energy released and incremental vacuole concentration versus vacuole fraction have been given to show how dependent the predictions are to the particle debonding phenomenon. The data from all particulate composites tested in the experimental portion of the study are shown here.

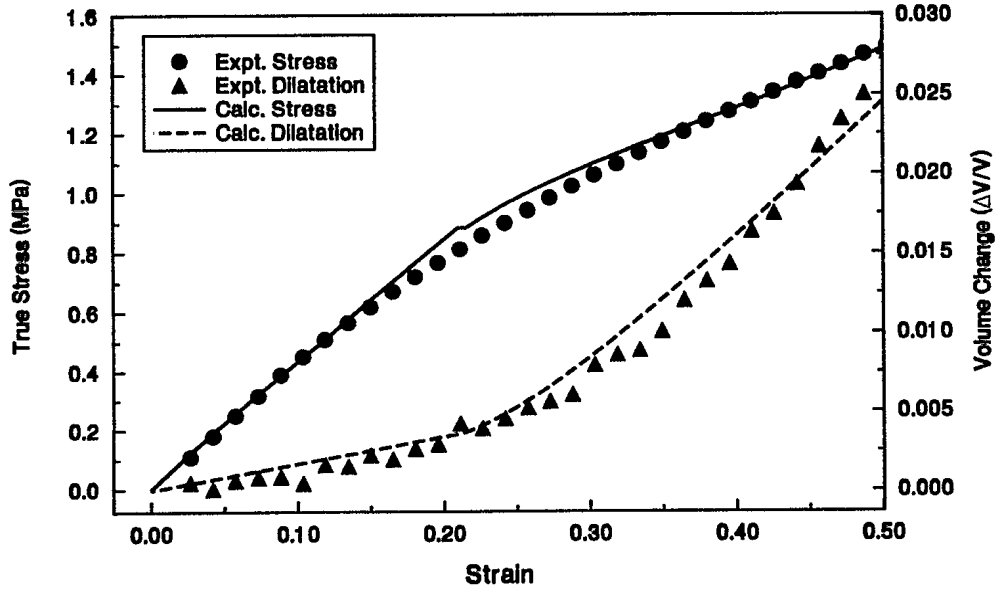

FIGURE C1 – True stress and dilatation behavior for composite T3FS

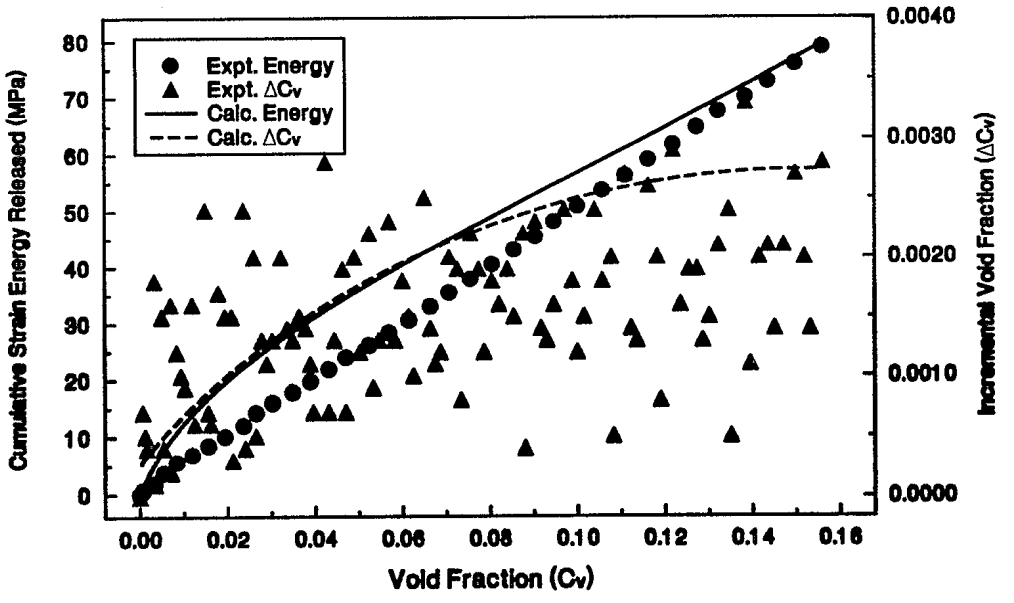

FIGURE C2 – Calculated strain energy and incremental vacuole concentration behavior for composite T3FS

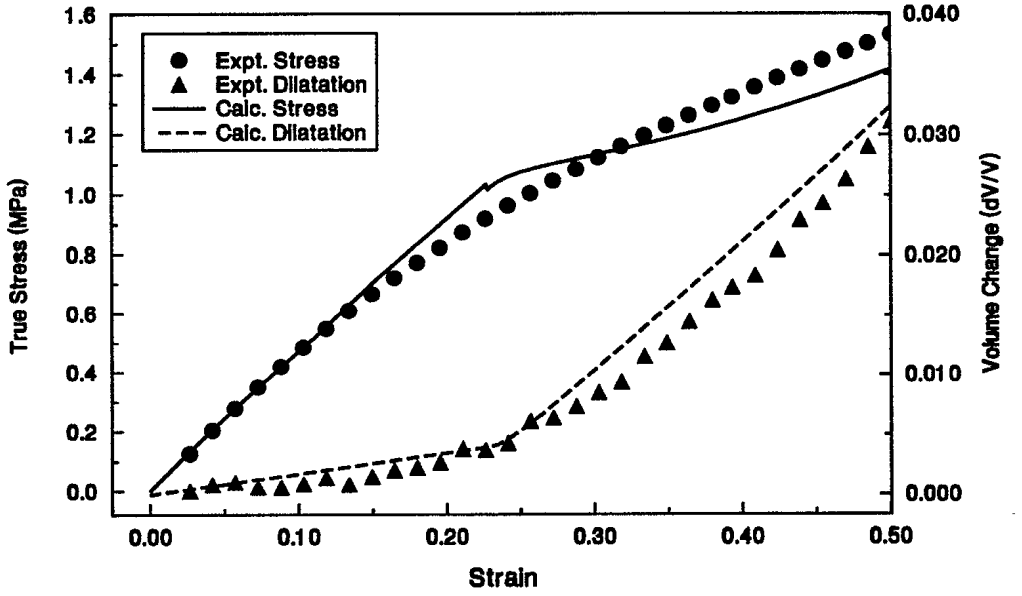

FIGURE C3 – True stress and dilatation behavior for composite T3FL

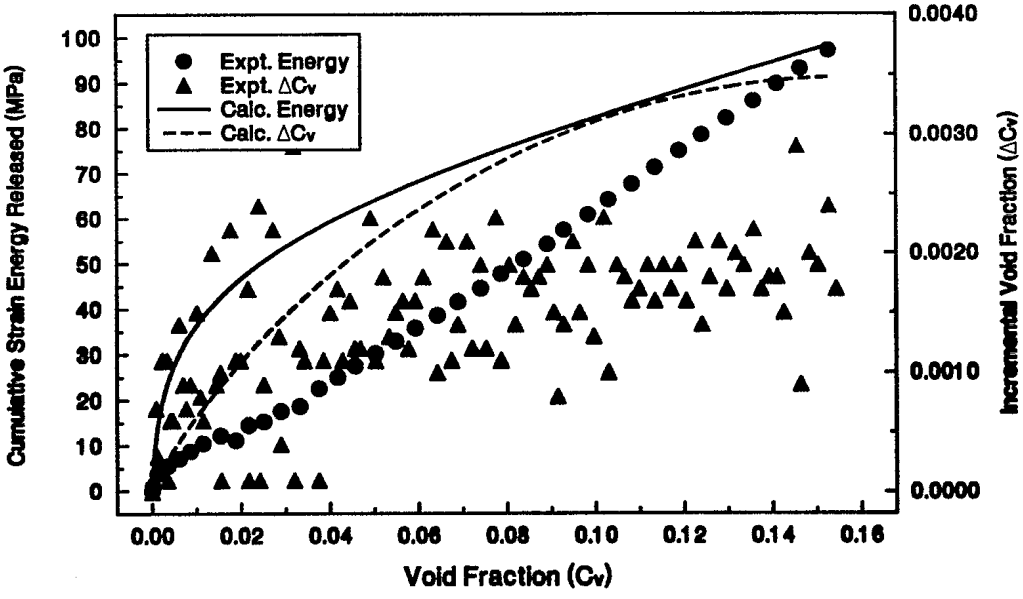

FIGURE C4 – Calculated strain energy and incremental vacuole concentration behavior for composite T3FL

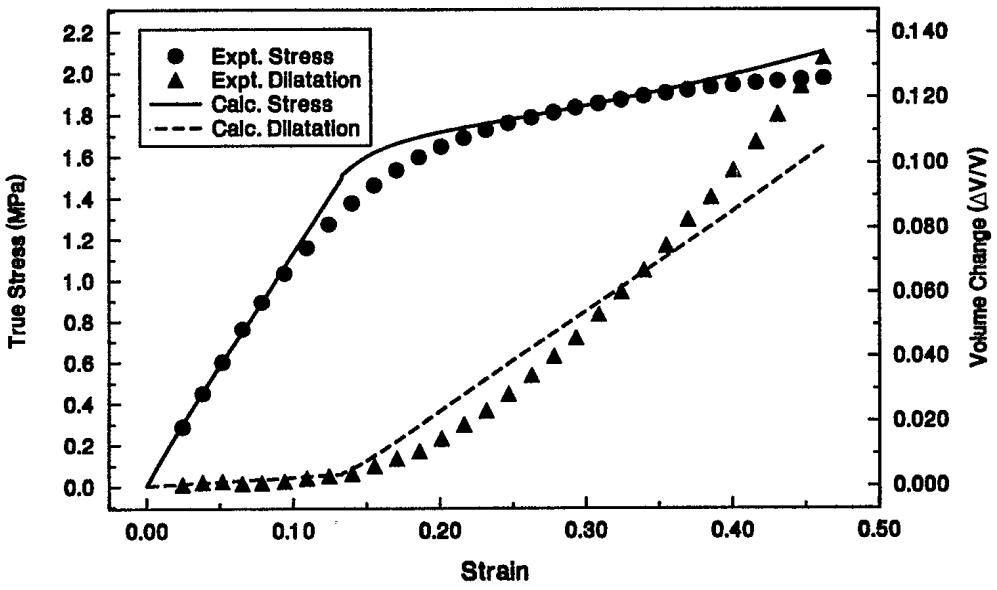

FIGURE C5 – True stress and dilatation behavior for composite T5FS

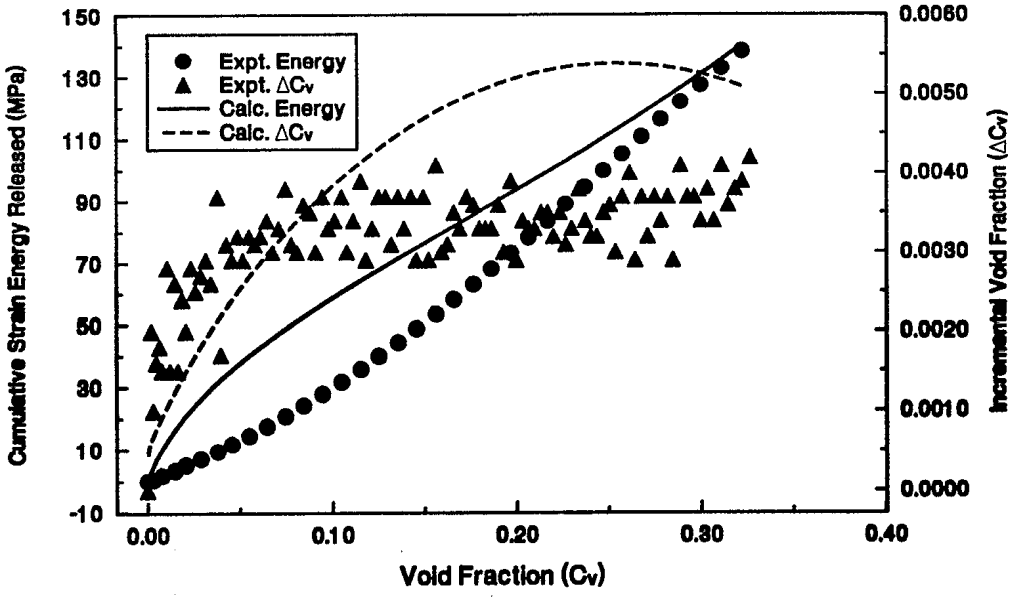

FIGURE C6 – Calculated strain energy and incremental vacuole concentration behavior for composite T5FS

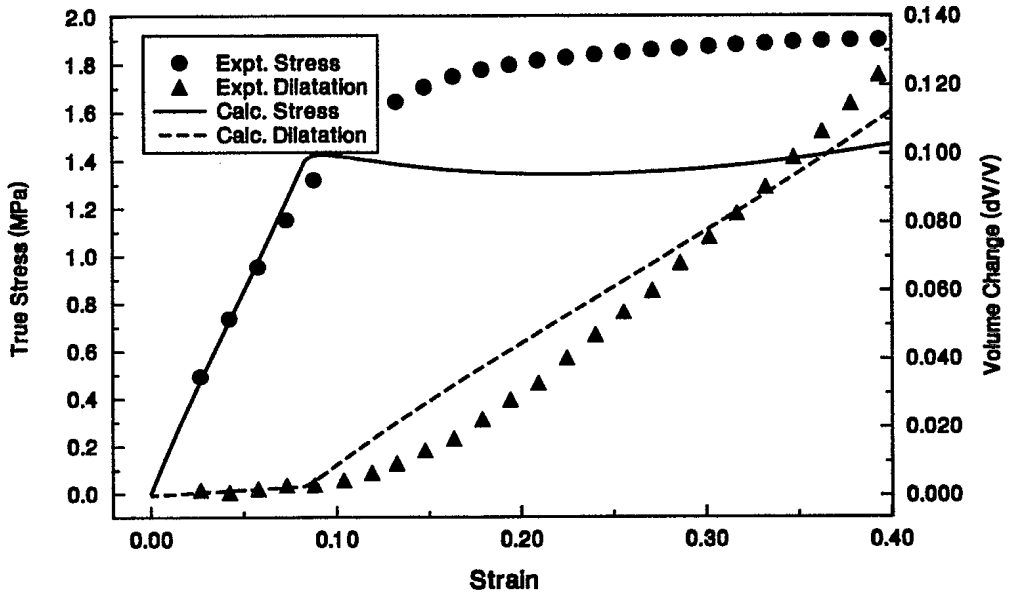

FIGURE C7 – True stress and dilatation behavior for composite T5FL

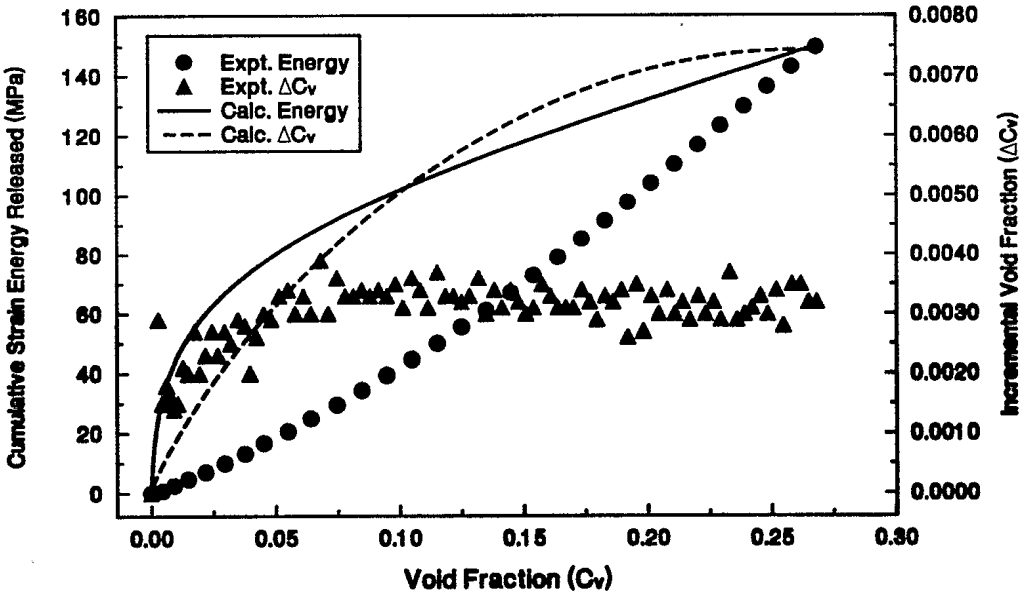

FIGURE C8 – Calculated strain energy and incremental vacuole concentration behavior for composite T5FL

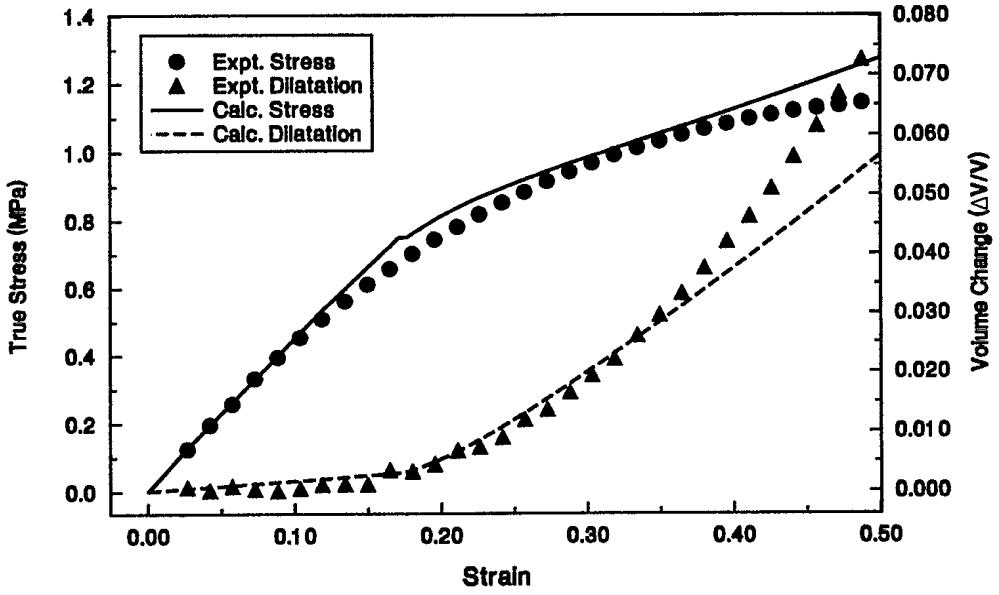

FIGURE C9 – True stress and dilatation behavior for composite N3FS

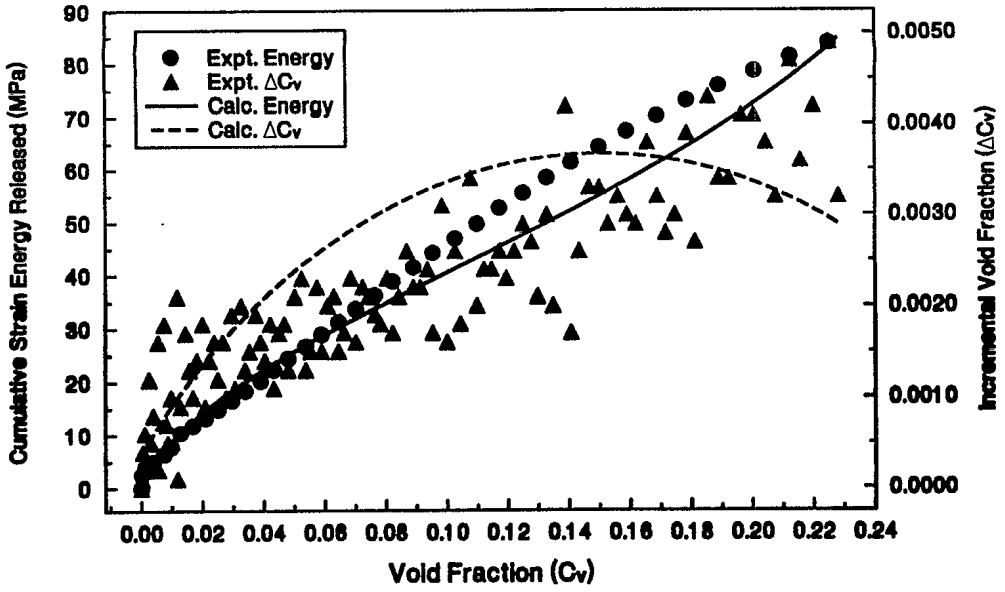

FIGURE C10 – Calculated strain energy and incremental vacuole concentration behavior for composite N3FS

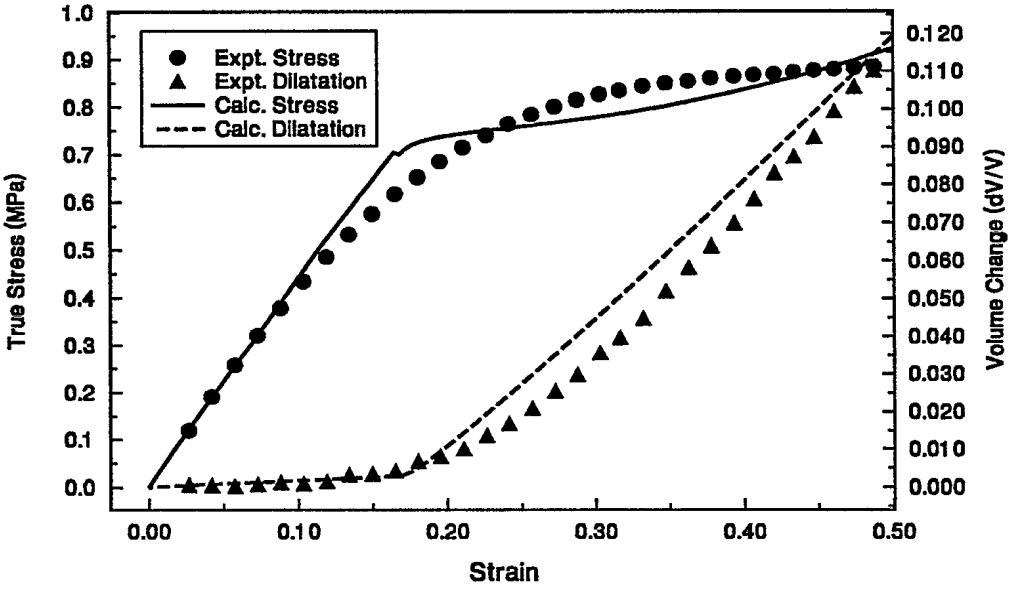

FIGURE C11 – True stress and dilatation behavior for composite N3FL

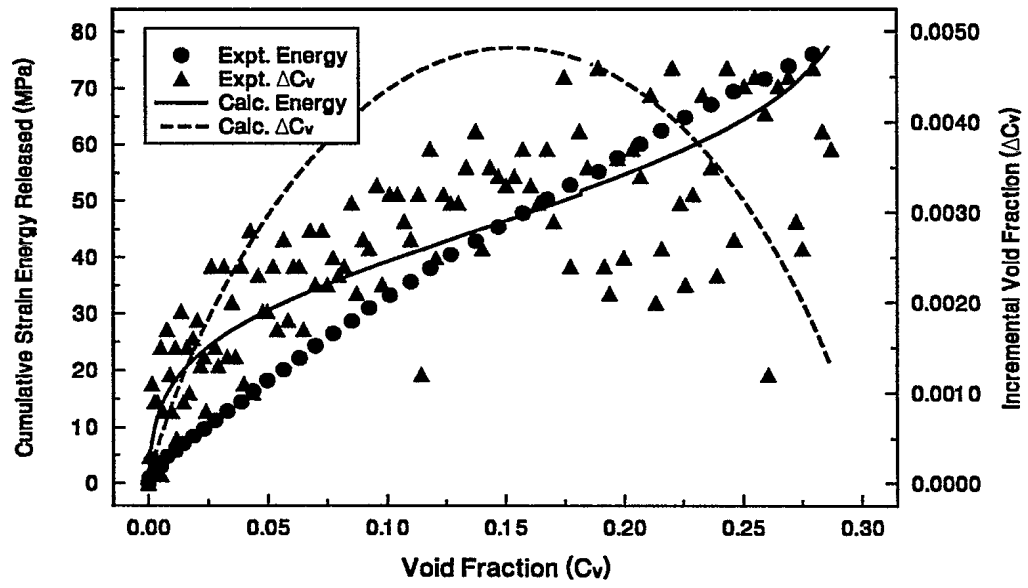

FIGURE C12 – Calculated strain energy and incremental vacuole concentration behavior for composite N3FL

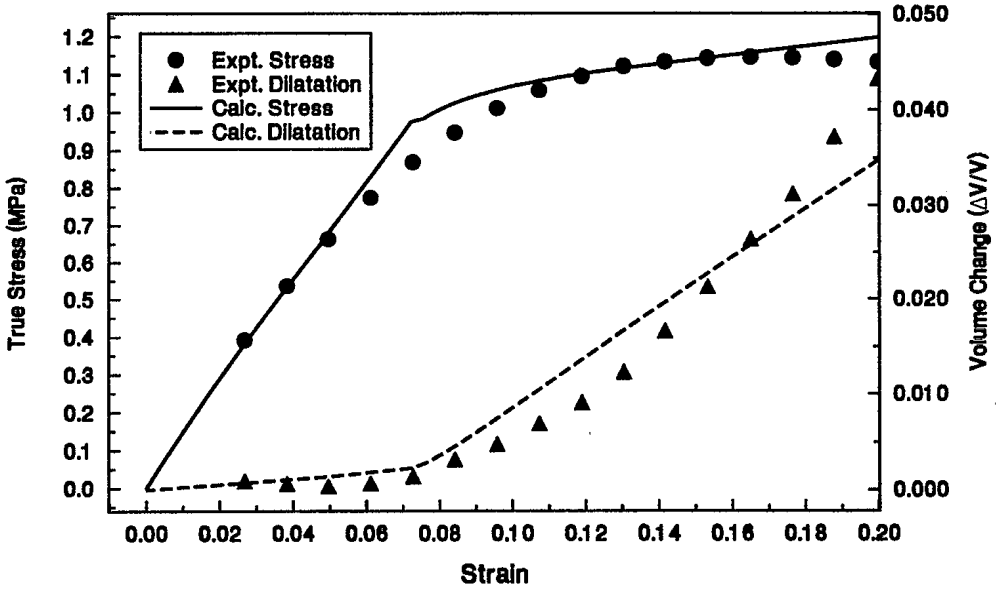

FIGURE C13 – True stress and dilatation behavior for composite N5FS

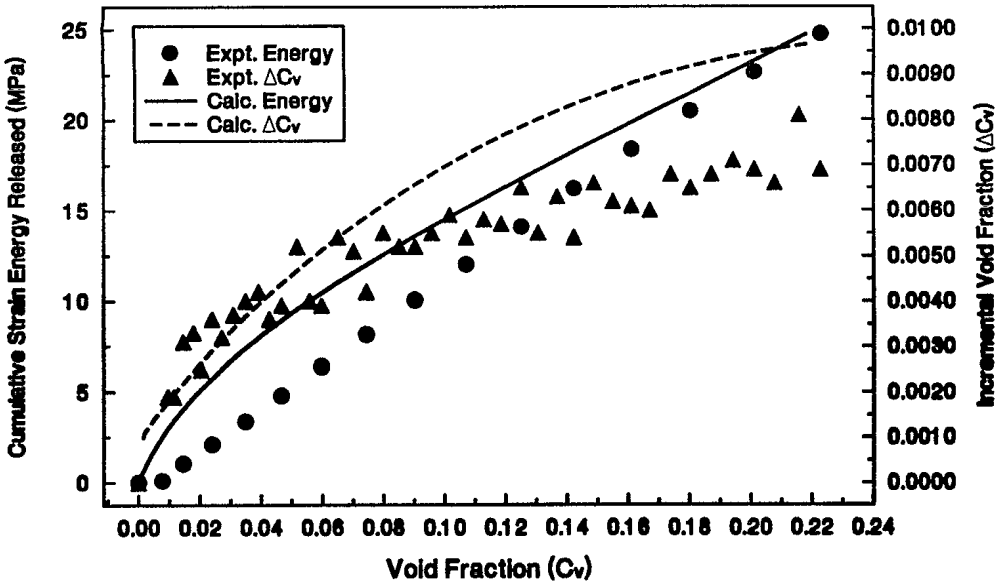

FIGURE C14 – Calculated strain energy and incremental vacuole concentration behavior for composite N5FS

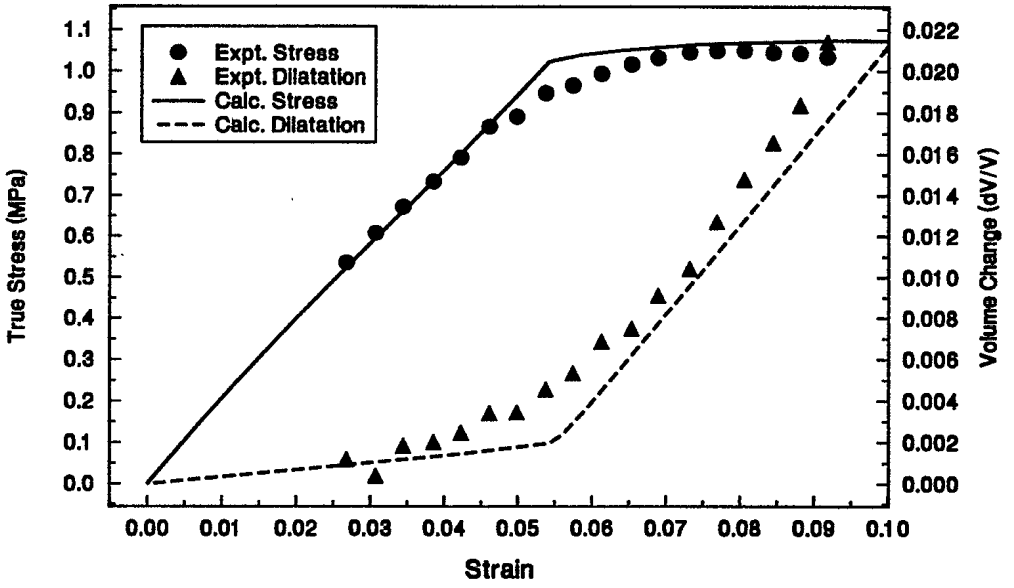

FIGURE C15 – True stress and dilatation behavior for composite N5FL

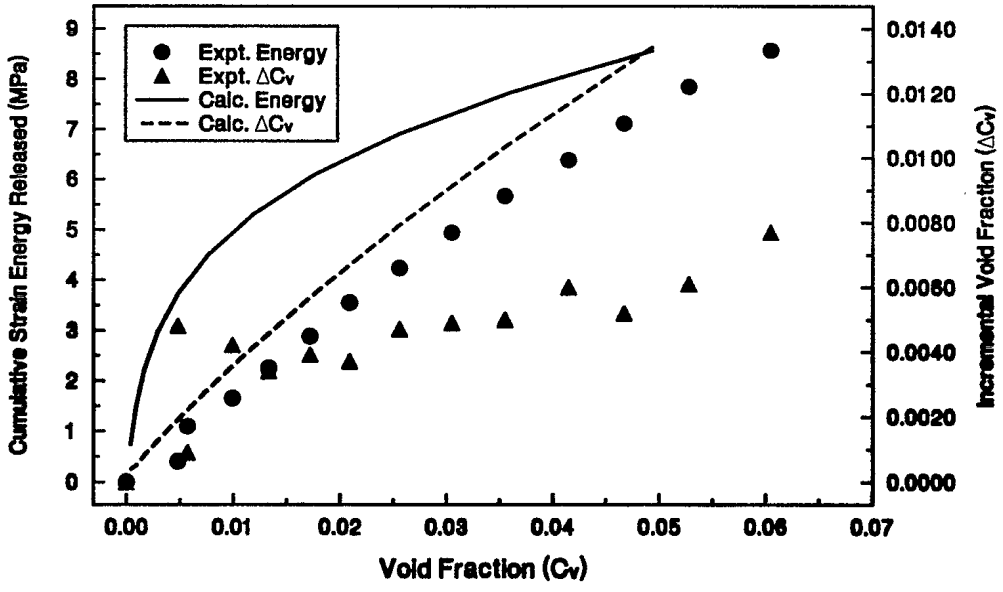

FIGURE C16 – Calculated strain energy and incremental vacuole concentration behavior for composite N5FL

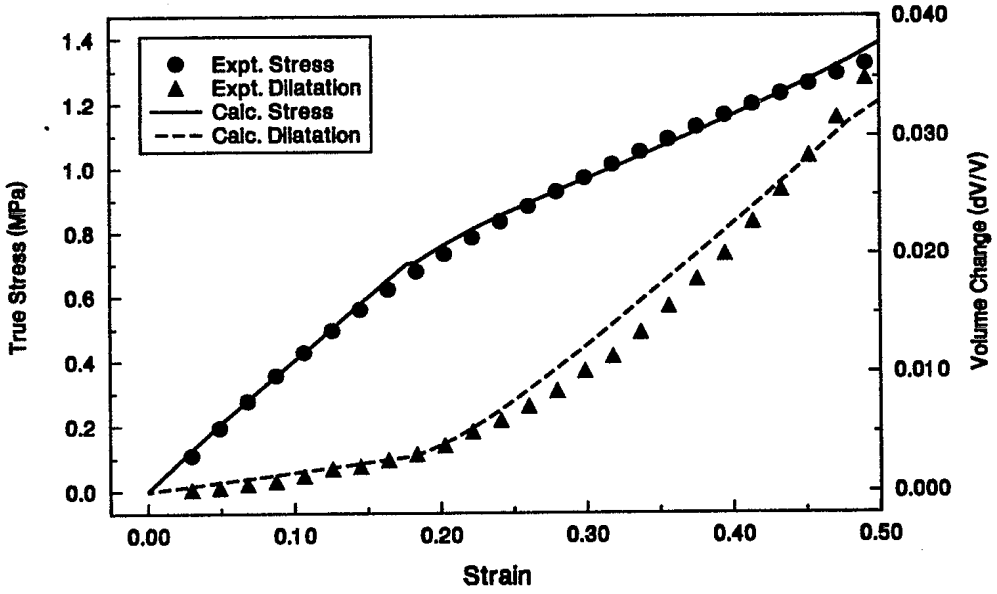

FIGURE C17 – True stress and dilatation behavior for composite T3MS

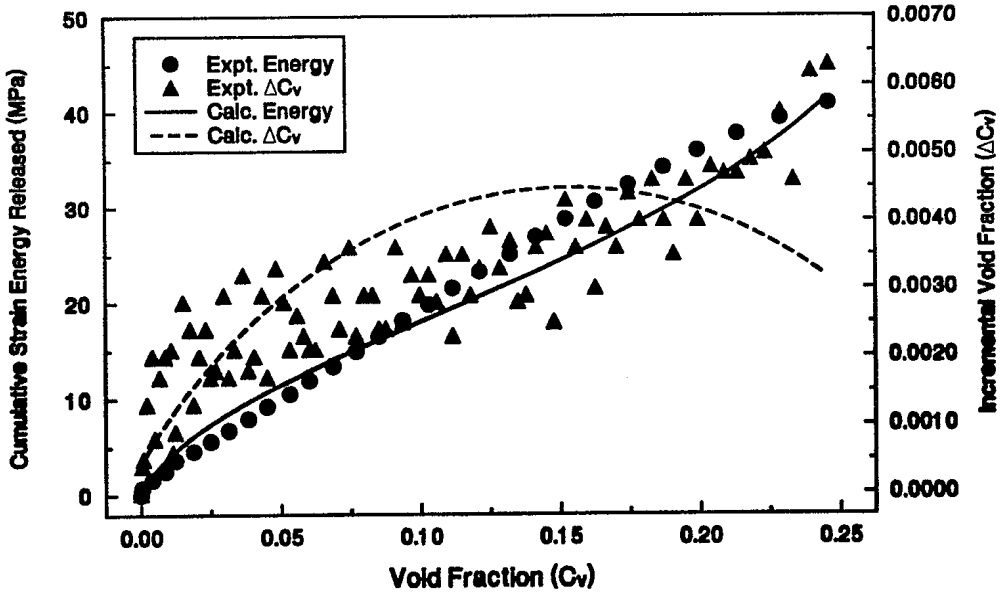

FIGURE C18 – Calculated strain energy and incremental vacuole concentration behavior for composite T3MS

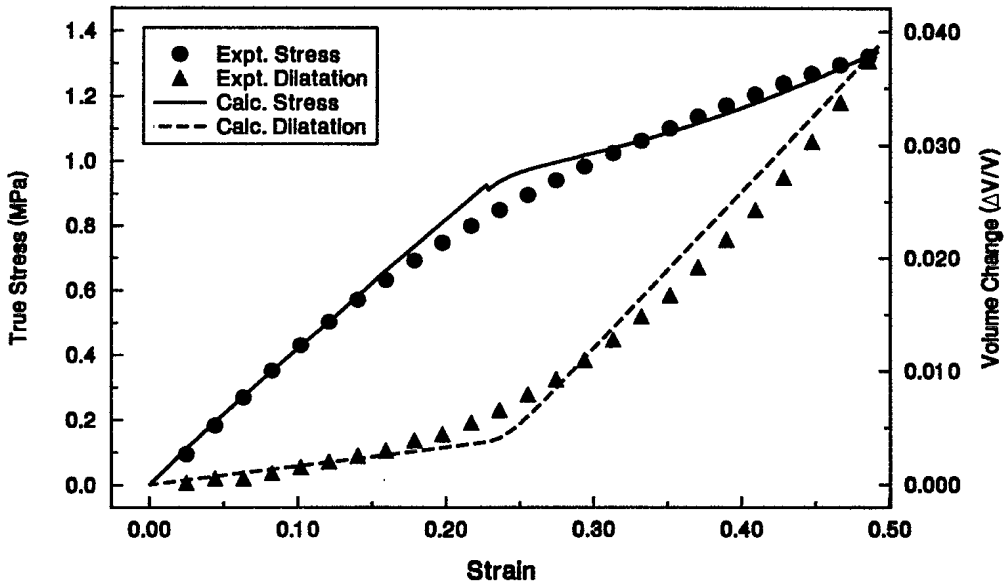

FIGURE C19 – True stress and dilatation behavior for composite T3ML

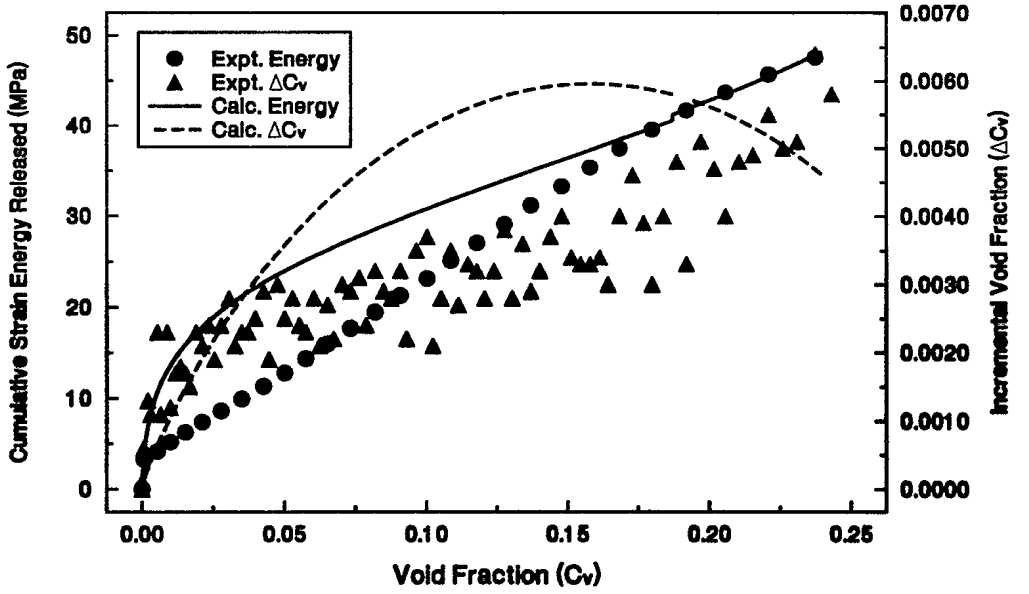

FIGURE C20 – Calculated strain energy and incremental vacuole concentration behavior for composite T3ML

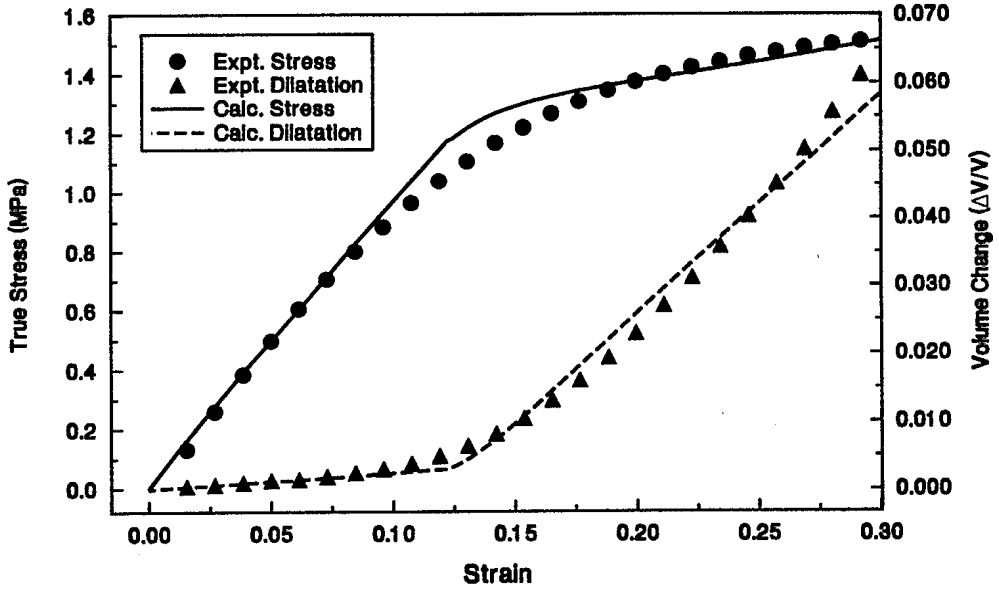

FIGURE C21 – True stress and dilatation behavior for composite T5MS

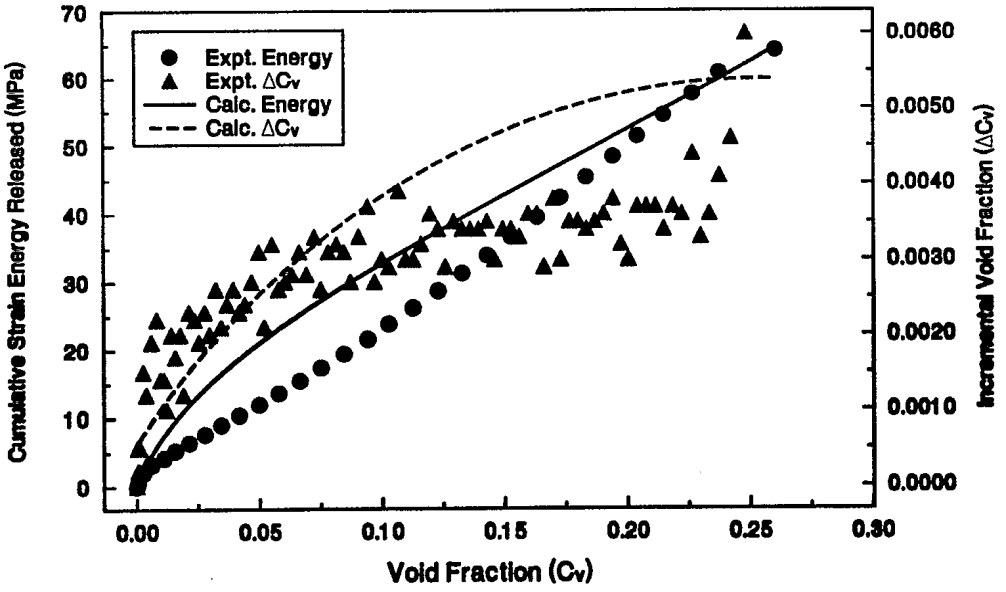

FIGURE C22 – Calculated strain energy and incremental vacuole concentration behavior for composite T5MS

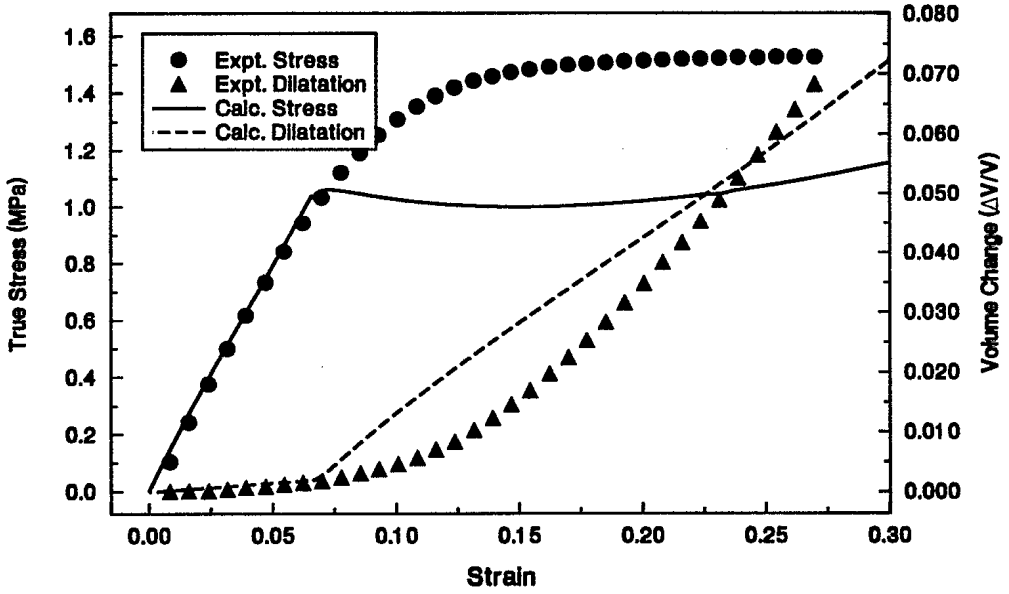

FIGURE C23 – True stress and dilatation behavior for composite T5ML

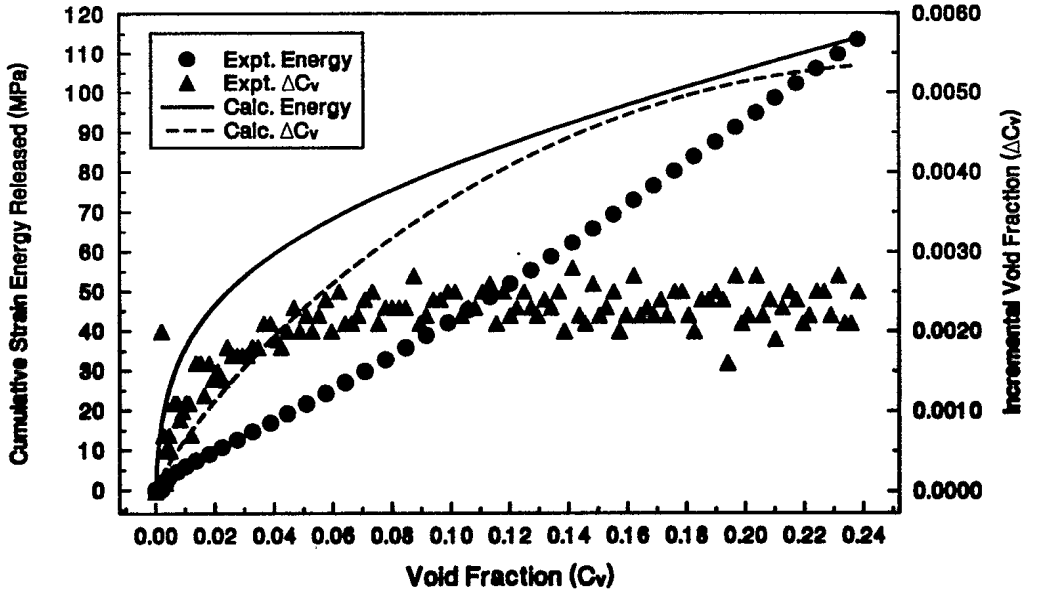

FIGURE C24 – Calculated strain energy and incremental vacuole concentration behavior for composite T5ML

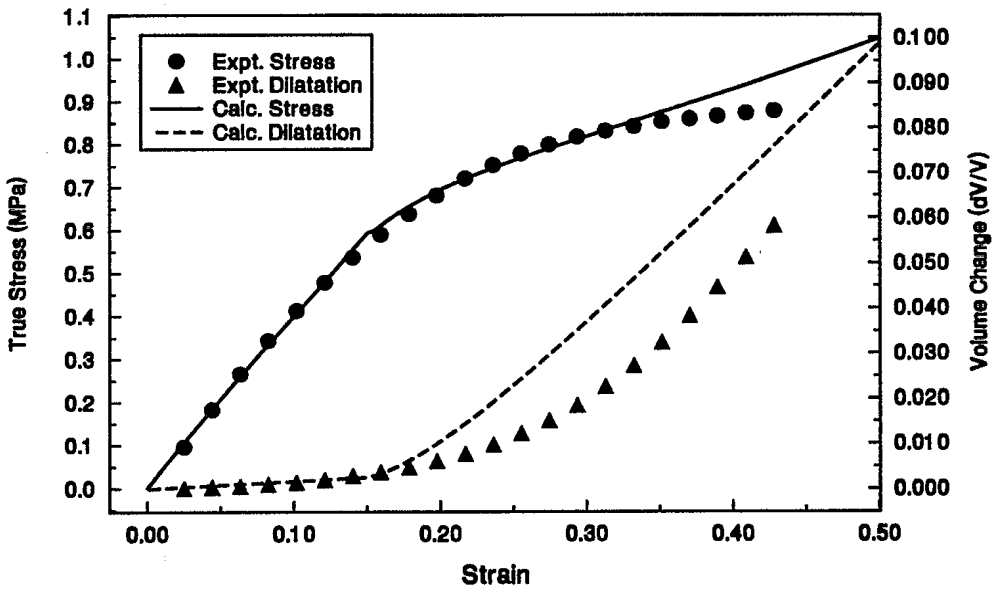

FIGURE C25 – True stress and dilatation behavior for composite N3MS

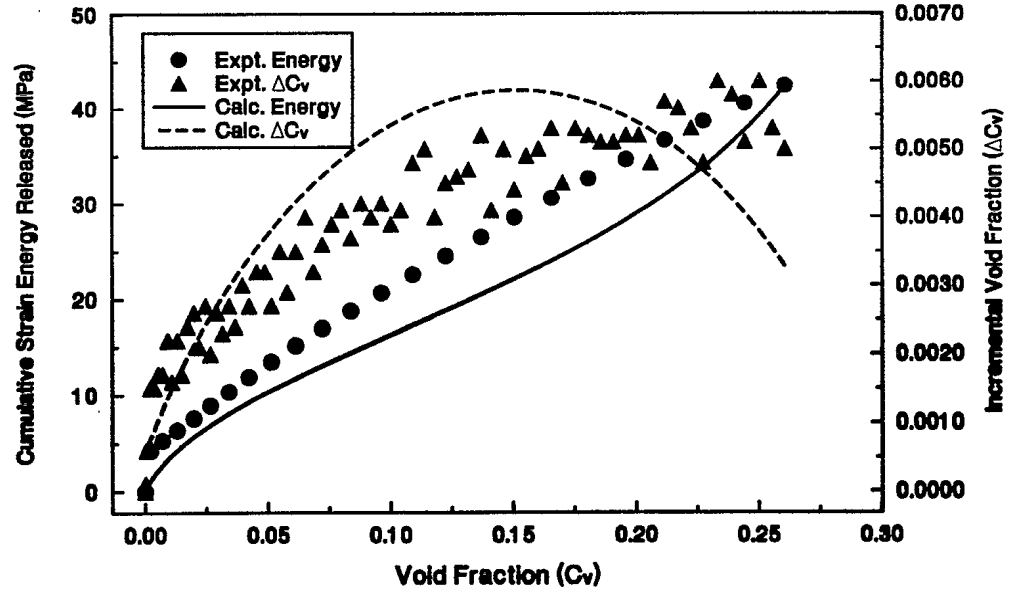

FIGURE C26 – Calculated strain energy and incremental vacuole concentration behavior for composite N3MS

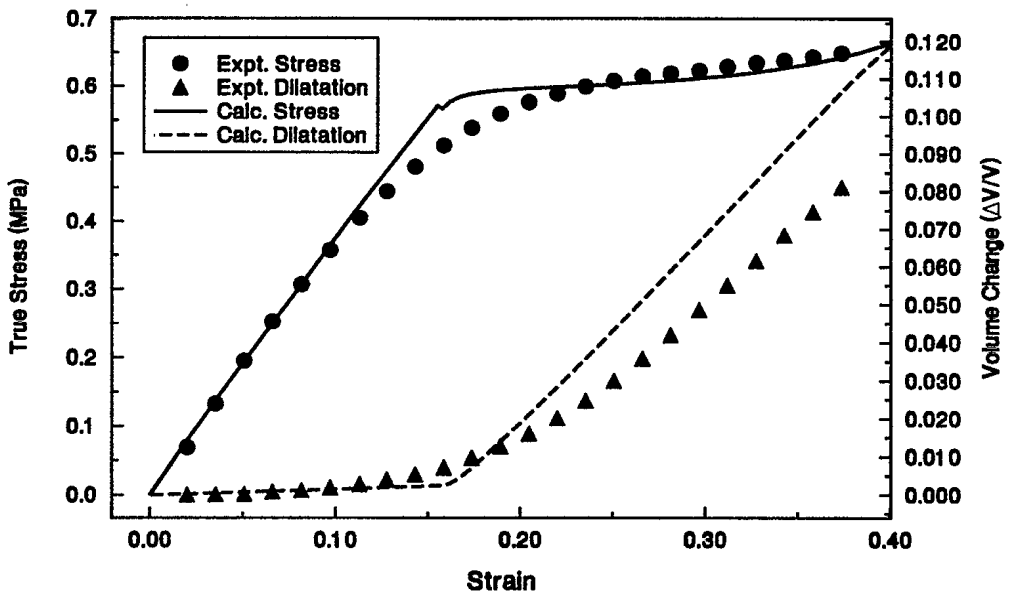

FIGURE C27 – True stress and dilatation behavior for composite N3ML

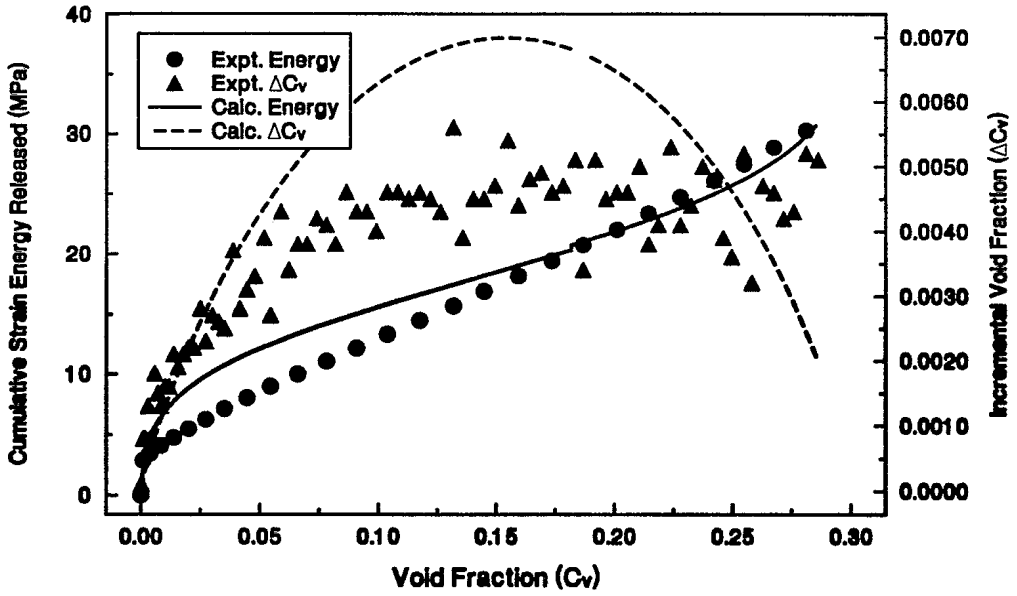

FIGURE C28 – Calculated strain energy and incremental vacuole concentration behavior for composite N3ML

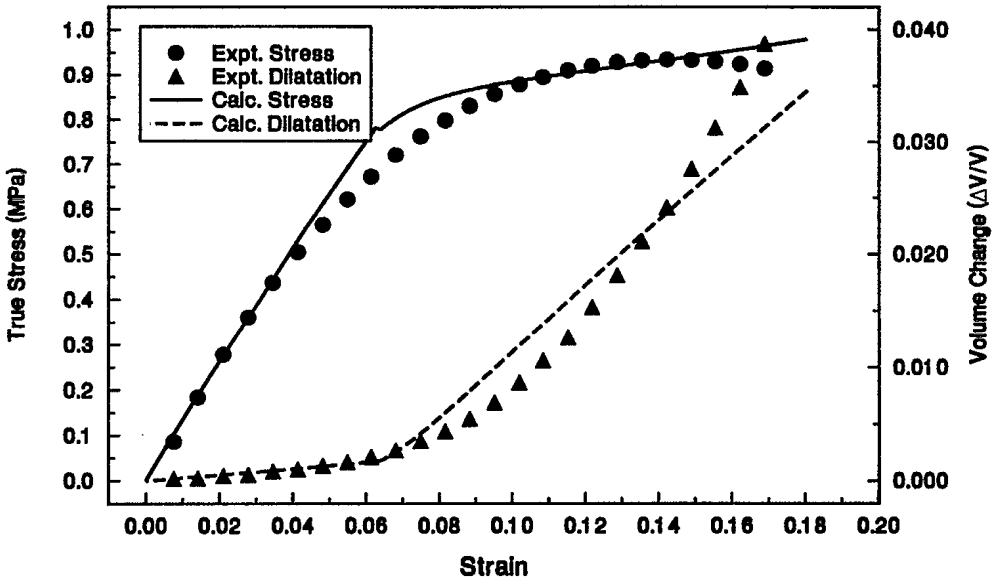

FIGURE C29 – True stress and dilatation behavior for composite N5MS

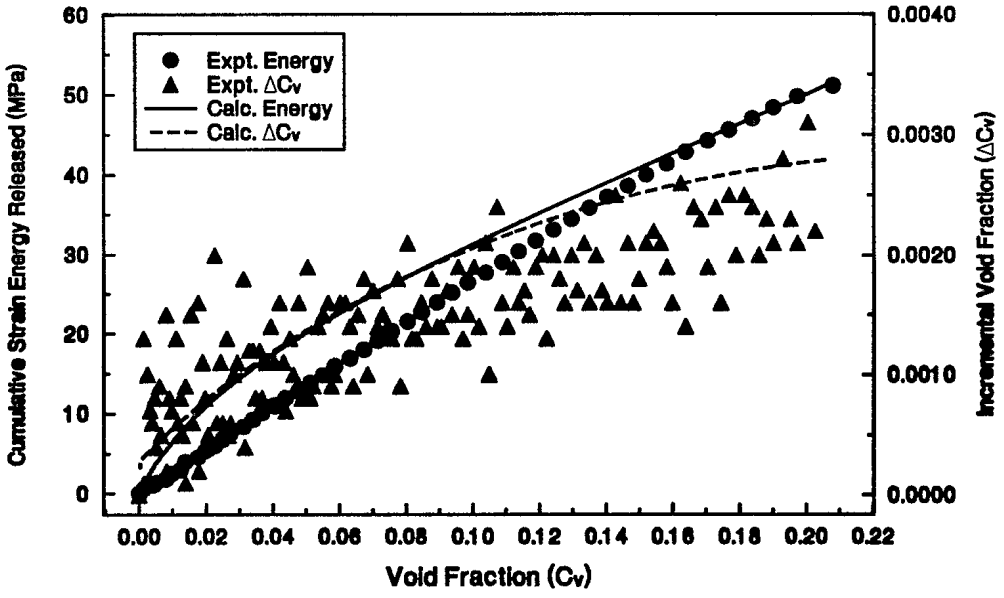

FIGURE C30 – Calculated strain energy and incremental vacuole concentration behavior for composite N5MS

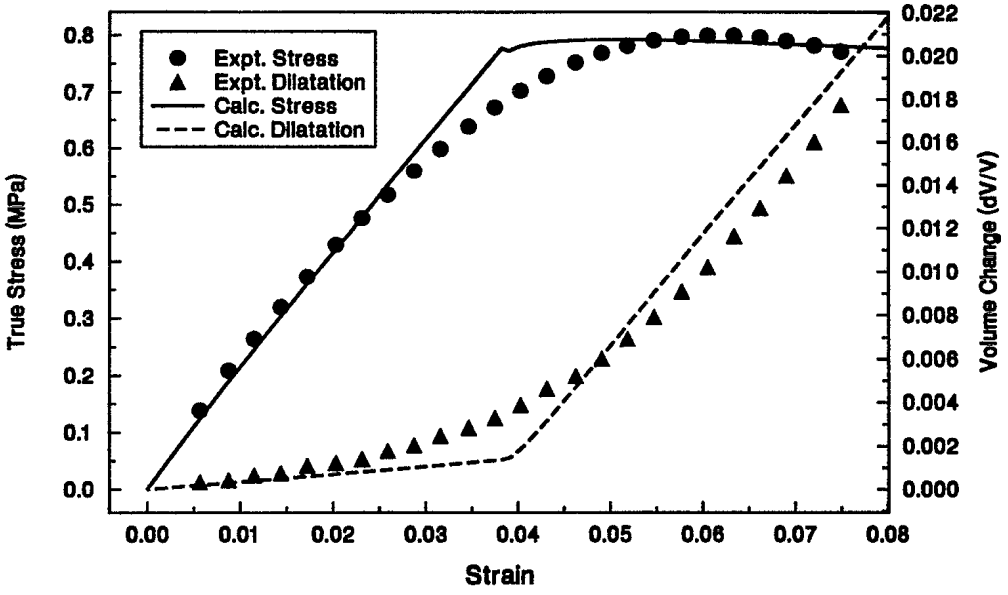

FIGURE C31 – True stress and dilatation behavior for composite N5ML

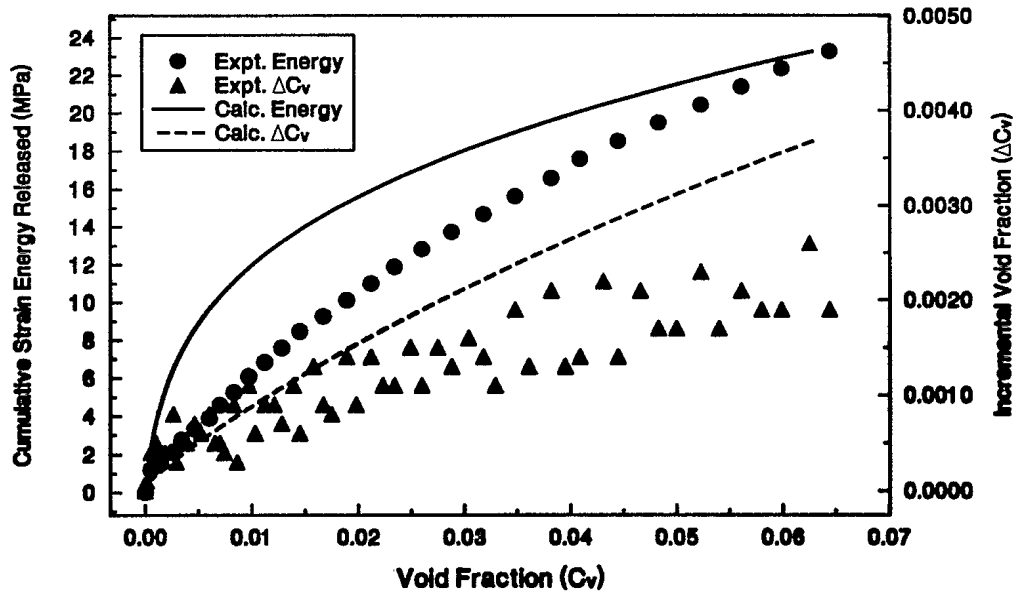

FIGURE C32 – Calculated strain energy and incremental vacuole concentration behavior for composite N5ML

APPENDIX D

FORTRAN PROGRAM FOR SENSITIVITY ANALYSIS

Sample '-14' Input File

```
ct3fl, part intact, vacuole          ! information header
220                                  ! no. pts. in curve
1                                    ! no. distributions
65.0                                 ! avg. particle radius (micron)
0.0374                               ! log std. dev.
0.311,0.0                            ! inclusion and void fraction
4609                                 ! sample volume (mm3)
2.1E-4,1.69,0,1,0                    ! Fb,Ym,w-type,m-type,v-type
0.4363E6,30E9                        ! matrix and inclusion shear modulus (Pa)
0.495,0.16                           ! matrix and inclusion Poisson ratio
30E9,34.12E9                         ! matrix and inclusion bulk modulus (Pa)
5.720,0                              ! Gc, pressure
2,1E-3,20                            ! a-type, conv. tol., max. iterations
10,0.5E6,0.0,2                     !no.steps init.,crit.strs,dbnd angle,avg.fac.
1.522255E6,-0.460286E6,0.270235E6    ! nonlinear co-effs in Young's modulus (Pa)
0.1,1,0.008,4        (NOT USED)      ! Ym step,Ym max iter,Ym %err,start index
0.0001,0.005,1       (NOT USED)      ! Cv step,Cv %err,start index
0.1E-4,1,0.001       (NOT USED)      ! Fb step,Fb max iter,Fb abs.err
```

Sample '-s' Input File

```
ct3ml                              ! information header
1,0.3995,1.188,0.02298           ! analysis type,target strain,stress,dilatation
-0.25  -0.25                       ! Ym range, Fb range
-0.20  -0.20                       !    "          "
-0.15  -0.15
-0.10  -0.10
-0.05  -0.05
 0.05   0.05
 0.10   0.10
 0.15   0.15
 0.20   0.20
 0.25   0.25
```

FORTRAN Listing for P14S.FOR

```
C==== main program
C      P14S.FOR
C====
C     used to carry out sensitivity analysis of Ym and Fb or crit strs
C     and Gc. based on P14 with additions to cycle through various Ym
C     and Fb. took out various P14 subroutines that are not needed for
C     this kind of analysis.
C     program reads "-14" file for optimal Ym and Fb or crit strs and Gc
C     values as well as values of properties etc. also needs to read an
C     "-s" file that specifies the strain level for results and range of
C     Ym and Fb or crit strs and Gc to be examined. the "-s" file must
C     have the following format:
C        information header
C        type, target strain, true stress, dilatation
C        column of ranges for Ym and Fb (up to 10)
C
C     type 1 - Ym and Fb, 2 - crit strs and Gc
C
C     last revision: 11 NOV 1996  11H15
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1,NSMAX = 10)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /MATRA/ BETA(2),WI(3,3),WV(3,3),IDENT(3,3)
      COMMON /MATRB/ S(3,3),CA(3,3),CB(3,3),CE(3,3),CF(3,3)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
```

```
     *    C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
          COMMON /PROPC/ AK0(1),AK1(1),AK2(1),STRNP1(NPTMX)
          COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *    PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *    IPDIST(NPTMX)
          COMMON /SENS/ YMRANG(NSMAX),FBRANG(NSMAX),STRLIM(5),
     *    STRDAT(NSMAX,NSMAX),DILDAT(NSMAX,NSMAX)
          CHARACTER FILNM*5
C
C==   initialize variables and arrays by BLOCK DATA INIT
C
          CALL INPUT(NDIST,NTOT,VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS,GAMM,
     *    PRESS,FILNM,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,DBANG,ISKIP)
C
          IABORT = 0
C==   cycle through range of Ym (X) and Fb (Y)
          YMOPT = YMULT
          FBOPT = FDBND
          STSOPT = STSFUD
          GCOPT = GAMM
          ITYP = INT(STRLIM(5))
          DO 20 ISEN = 1,INT(STRLIM(2))
             IF (ITYP.EQ.1) THEN
                YMULT = YMRANG(ISEN)*YMOPT+YMOPT
             ELSE
                STSFUD = YMRANG(ISEN)*STSOPT+STSOPT
             ENDIF
             DO 10 JSEN = 1,INT(STRLIM(2))
                IF (ITYP.EQ.1) THEN
                   FDBND = FBRANG(JSEN)*FBOPT+FBOPT
                   WRITE (6,'(A,E11.4,A,E11.4)') ' Ym = ',YMULT,' Fb = ',
     *                FDBND
                ELSE
                   GAMM = FBRANG(JSEN)*GCOPT+GCOPT
                   WRITE (6,'(A,E11.4,A,E11.4)') ' crit.strs = ',STSFUD,
     *                ' Gc = ',GAMM
                ENDIF
```

```
C
            CALL STRSTN(NDIST,NTOT,NPTS,VOLSMP,FDBND,YMULT,IKIND,IMORI,
     *          IPOIS,GAMM,PRESS,DILATO,IWRT,IAUG,STNTOL,ITERMX,NLTOT,
     *          STSFUD,STRNP,DBANG,ISKIP,ISEN,JSEN,IABORT)
C
   10    CONTINUE
   20 CONTINUE
      YMULT = YMOPT
      FDBND = FBOPT
      STSFUD = STSOPT
      GAMM = GCOPT
C
C==   write out data
      CALL STRWRT(NDIST,NPTS,VOLSMP,GAMM,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *   PRESS,DILATO,FILNM,NLTOT,STSFUD,IWRT,DBANG,ISKIP,IABORT)
C
      END
C
C
      SUBROUTINE INPUT(NDIST,NTOT,VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *   GAMM,PRESS,FILNM,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,DBANG,
     *   ISKIP)
C==== reads in problem input either by file or keyboard. if data entered
C     through a file, user inputs name only, a file extension of DAT is
C     assumed. the first line in the input file is used for a user
C     heading and is not read in, constituent material properties
C     designated as follows: 1-inclusion,2-void or vacuole,3-matrix
C     STRNLM (1)=strain of interest (2)=no.of ranges (3)=expt.stress
C     (4)=expt.dilat
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1,NSMAX = 10)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
```

```
*    VLFRVO(NTDIS)
     COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
     COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
     COMMON /SENS/ YMRANG(NSMAX),FBRANG(NSMAX),STRLIM(5),
*    STRDAT(NSMAX,NSMAX),DILDAT(NSMAX,NSMAX)
     CHARACTER ANS*1,FILNM*5
C
     WRITE (6,'(A)') ' File to read? (-14.INP will be appended)'
     READ (5,'(A5)') FILNM
     WRITE (6,'(/,A,A5,A)') ' Reading ',FILNM,'-14.INP'
     OPEN (UNIT=7,FILE=FILNM//'-14.INP',FORM='FORMATTED',STATUS='OLD')
     READ (7,*)
     READ (7,*) NTOT
     READ (7,*) NDIST
     DO 10 I = 1,ABS(NDIST)
        READ (7,*) RADAVG(I)
        READ (7,*) LOGSTD(I)
        READ (7,*) VLFRFO(I),VLFRVO(I)
  10 CONTINUE
     READ (7,*) VOLSMP
     READ (7,*) FDBND,YMULT,IKIND,IMORI,IPOIS
     READ (7,*) G(3),G(1)
     READ (7,*) POIS(3),POIS(1)
     READ (7,*) G(2),K(2)
     READ (7,*) GAMM,PRESS
     READ (7,*) IAUG,STNTOL,ITERMX
     READ (7,*) NLTOT,STSFUD,DBANG,ISKIP
     READ (7,*) AKO(1),AK1(1),AK2(1)
     CLOSE (7)
C
     WRITE (6,'(A,A5,A)') ' Reading ',FILNM,'-S.INP'
     ICNT = 0
     OPEN (UNIT=7,FILE=FILNM//'-S.INP',FORM='FORMATTED',STATUS='OLD')
     READ (7,*)
     READ (7,*) STRLIM(5),STRLIM(1),STRLIM(3),STRLIM(4)
     DO 20 I = 1,NSMAX
        READ (7,*,END=30) YMRANG(I),FBRANG(I)
```

```
         ICNT = ICNT+1
   20 CONTINUE
   30 CONTINUE
      STRLIM(2) = REAL(ICNT)
C
C==   set write file flag, 0=STRWRT, 1=STRWRT,DBGWRT,HSTWRT, 2=all
      IWRT = 0
      NDIST = ABS(NDIST)
      NTOT = ABS(NTOT)
      IF (IAUG.EQ.0) NLTOT = 0
C
      RETURN
      END
C
C
      SUBROUTINE STRSTN(NDIST,NTOT,NPTS,VOLSMP,FDBND,YMULT,IKIND,IMORI,
     *   IPOIS,GAMM,PRESS,DILATO,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,
     *   STRNP,DBANG,ISKIP,ISEN,JSEN,IABORT)
C==== main subroutine which organizes particle size distribution,
C     composite property, critical strain and true stress and dilation
C     calculation modules.
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1,NSMAX = 10)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      COMMON /SENS/ YMRANG(NSMAX),FBRANG(NSMAX),STRLIM(5),
     *   STRDAT(NSMAX,NSMAX),DILDAT(NSMAX,NSMAX)
```

```
C
C==   initialize abort flag
      IABORT = 0
C
C     IF(IWRT.EQ.0) THEN
C        WRITE (6,'(/,A)') ' Generating particle distribution'
      CALL GAUSS(NDIST,NTOT,NPTS,IABORT)
C        WRITE (6,'(A)') ' Finding particle size and number'
      CALL PARTSZ(NDIST,NPTS,VOLSMP,IABORT)
C        WRITE (6,'(A)') ' Sorting particle distributions'
      CALL SORTER(NDIST,NPTS,IABORT)
C        WRITE (6,'(A)') ' Calculating vol frac and dA/dc'
      CALL VOLFRC(NDIST,NPTS,VOLSMP,DBANG,IABORT)
C        IWRT=1
C     ENDIF
C
      WRITE (6,'(A)') ' Generating true stress-strain curve'
C==   calculate initial composite properties
      ICNT = 1
      CONCI = NETVF(ICNT)
      CONCV = NETVV(ICNT)
      STRNO = 0
      CALL MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,IPOIS,STRNO,
     *   IAUG,IABORT)
C
      CALL NLSTRS(NDIST,NPTS,FDBND,YMULT,IKIND,IMORI,IPOIS,IAUG,STNTOL,
     *   ITERMX,NLTOT,STSFUD,STRNP,DILATO,IABORT)
C
C==   main routine for debonding and true stress-strain calculation.
C     offset pointer ICNT by 1 to make room for undebonded state.
C     STRNP in NLSTRS is matrix strain at end of initial portion.
C     matrix strain revised if e_i > e_max, if it is not STRNP
C     maintained at current e_max. IPDIST used as a flag to show when
C     STRNP has been updated.
C
      CRTMAX = CRTSTN(NLTOT+1)
      DO 10 ICNT = 2+NLTOT,NDIST*NPTS+NLTOT+1
```

```
          CONCI = NETVF(ICNT)
          CONCV = NETVV(ICNT)
C
          CALL MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *       STRNP,IAUG,IABORT)
          CALL CRIT(ICNT,NLTOT,IAUG,VOLSMP,GAMM,PRESS,CRTMAX,STRNP,ISKIP,
     *       IABORT)
          STRNP1(ICNT) = STRNP
          CALL CALVAL(ICNT,PRESS,DILATO,IABORT)
C
C     store sensitivity info in matrices
          IF (CRTSTN(ICNT).GE.0.97*STRLIM(1)) THEN
             STRDAT(ISEN,JSEN) = STRESS(ICNT)
             DILDAT(ISEN,JSEN) = DILAT(ICNT)
             RETURN
          ENDIF
C
   10 CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE GAUSS(NDIST,NTOT,NPTS,IABORT)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
      FUNCTION FUNC(X)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE PARTSZ(NDIST,NPTS,VOLSMP,IABORT)
       (subroutine details may be found in Appendix E)
      RETURN
```

```
      END
C
C
      SUBROUTINE SORTER(NDIST,NPTS,IABORT)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE VOLFRC(NDIST,NPTS,VOLSMP,DBANG,IABORT)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *   STNTMP,IAUG,IABORT)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE CRIT(ICNT,NLTOT,IAUG,VOLSMP,GAMM,PRESS,CRTMAX,STRNP,
     *   ISKIP,IABORT)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE CALVAL(ICNT,PRESS,DILATO,IABORT)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE NLSTRS(NDIST,NPTS,FDBND,YMULT,IKIND,IMORI,IPOIS,IAUG,
     *   STNTOL,ITERMX,NLTOT,STSFUD,STRNP,DILATO,IABORT)
```

```
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      BLOCK DATA INIT
C==== initialize all variables and arrays used in program
C     check NPTMX if NTDIS or GSMX are changed. check matrices
C     STRDAT and DILDAT if NSMAX is changed.
C        NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1,NSMAX = 10)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /MATRA/ BETA(2),WI(3,3),WV(3,3),IDENT(3,3)
      COMMON /MATRB/ S(3,3),CA(3,3),CB(3,3),CE(3,3),CF(3,3)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      COMMON /SENS/ YMRANG(NSMAX),FBRANG(NSMAX),STRLIM(5),
     *   STRDAT(NSMAX,NSMAX),DILDAT(NSMAX,NSMAX)
C
      DATA Z /GSMX*0/ RADIUS /NPTMX*0/ PROB /NPTMX*0/
      DATA NUMPAR /NPTMX*0/ VOLPAR /NPTMX*0/ NETVF /NPTMX*0/ NETVV /
     *   NPTMX*0/ DADC /NPTMX*0/ NPARTL /NTDIS*0/
      DATA RADAVG /NTDIS*0/ LOGSTD /NTDIS*0/ VLFRFO /NTDIS*0/ VLFRVO /
     *   NTDIS*0/
```

```
      DATA BETA /2*0/ WI /9*0/ WV /9*0/ IDENT /1,0,0,0,1,0,0,0,1/
      DATA S /9*0/ CA /9*0/ CB /9*0/ CE /9*0/ CF /9*0/
      DATA K /3*0/ G /3*0/ E /3*0/ POIS /3*0/ CI /9*0/ CV /9*0/ CO /9*0/
      DATA C11 /NPTMX*0/ C12 /NPTMX*0/ C21 /NPTMX*0/ C22 /NPTMX*0/ C23 /
     *   NPTMX*0/ ECMP /NPTMX*0/ POISC /NPTMX*0/
      DATA AKO /1*0/ AK1 /1*0/ AK2 /1*0/ STRNP1 /NPTMX*0/
      DATA CRTSTN /NPTMX*0/ STRESS /NPTMX*0/ DILAT /NPTMX*0/ PRBSRV /
     *   NPTMX*0/ SORRAD /NPTMX*0/ SORPAR /NPTMX*0/ SORVLP /NPTMX*0/
     *   IPDIST /NPTMX*0/
      DATA YMRANG /NSMAX*0/ FBRANG /NSMAX*0/ STRLIM /5*0/ STRDAT /100*0/
     *   DILDAT /100*0/
C
      END
C
C
      SUBROUTINE QSIMP(FUNC,A,B,S)
       (subroutine details may be found in Appendix E)
      END
C
C
      SUBROUTINE TRAPZD(FUNC,A,B,S,N)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE SORT3(N,RA,RB,RC,IRD)
       (subroutine details may be found in Appendix E)
      END
C
C
      SUBROUTINE CALCIO(STNTMP,IAUG,IABORT)
       (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
```

```
      SUBROUTINE CALCCV(FDBND,IPOIS,IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE CMPRPO(IKIND,IMORI,IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
      SUBROUTINE CMPRP(CONCI,CONCV,YMULT,CAVG,IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE CALCW(IKIND,IMORI,IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE CALCS(IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE GAMMA(A,CONC,ITYPE,YMULT,IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE ADD(C,A,B)
        (subroutine details may be found in Appendix E)
```

```
      RETURN
      END
C
C
      SUBROUTINE SUB(C,A,B)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE MULT(C,A,B)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE INVERT(AI,A,IABORT)
        (subroutine details may be found in Appendix E)
      RETURN
      END
C
C
      SUBROUTINE STRWRT(NDIST,NPTS,VOLSMP,GAMM,FDBND,YMULT,IKIND,IMORI,
     *   IPOIS,PRESS,DILATO,FILNM,NLTOT,STSFUD,IWRT,DBANG,ISKIP,IABORT)
C==== write out true stress and dilatation results versus critical strain
C     include probability survival, radius, no. particles and
C     distribution info. write intermediate data where e_i<e_max if
C     IWRT>0.
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1,NSMAX = 10)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
```

```
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      COMMON /SENS/ YMRANG(NSMAX),FBRANG(NSMAX),STRLIM(5),
     *   STRDAT(NSMAX,NSMAX),DILDAT(NSMAX,NSMAX)
      CHARACTER FILNM*5
C
      IF (IABORT.EQ.1) RETURN
C
      WRITE (6,'(/,A,A5,A)') ' Writing to ',FILNM,'S.DAT'
C
      OPEN (UNIT=7,FILE=FILNM//'S.DAT',STATUS='UNKNOWN')
      WRITE (7,5000)
      DO 10 I = 1,NDIST
         WRITE (7,'(1X,I1,4(3X,0PE11.4))') I,RADAVG(I),LOGSTD(I),
     *      VLFRFO(I),VLFRVO(I)
   10 CONTINUE
C
      WRITE (7,5100) G(3),G(1),POIS(3),POIS(1),G(2),K(2)
      WRITE (7,5200) VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS
      WRITE (7,5300) PRESS,GAMM,DILATO,STSFUD,DBANG
      WRITE (7,5500) AKO(1),AK1(1),AK2(1),ISKIP
      IF (INT(STRLIM(5)).EQ.1) THEN
         WRITE (7,5400)
      ELSE
         WRITE (7,5700)
      ENDIF
      DO 30 ISEN = 1,INT(STRLIM(2))
         DO 20 JSEN = 1,INT(STRLIM(2))
C           STRFRC = STRDAT(ISEN,JSEN)
C           DILFRC = DILDAT(ISEN,JSEN)
            STRFRC = (STRDAT(ISEN,JSEN)/STRLIM(3))-1
            DILFRC = (DILDAT(ISEN,JSEN)/STRLIM(4))-1
```

```
            WRITE (7,5600) YMRANG(ISEN),FBRANG(JSEN),STRFRC,DILFRC
   20    CONTINUE
   30 CONTINUE
C
      CLOSE (7)
      RETURN
 5000 FORMAT (' #    avg Rad(um)    std dev          Vf           Vv')
 5100 FORMAT (' Gm(Pa)=',0PE11.4,' Gf(Pa)=',0PE11.4,' vm=',0PE11.4,
     *   ' vf=',0PE11.4,' Gv(Pa)=',0PE11.4,' Kv=',0PE11.4)
 5200 FORMAT (' V(mm3) =',0PE11.4,' frac dbnd=',0PE11.4,' Y-mult=',0PE11
     *   .4,' w-type=',I3,' m-type=',I3,' v-type=',I3)
 5300 FORMAT (' P0(Pa)=',0PE11.4,' Gc(Pa-m)=',0PE11.4,' (dV/V)0=',0PE11
     *   .4,' crit.strs(Pa)=',0PE11.4,' dbnd.ang(deg)=',0PE11.4)
 5400 FORMAT ('    Ym Frac      Fb frac     strs frac    dil frac')
 5500 FORMAT (' AK0(Pa)=',0PE11.4,' AK1(Pa)=',0PE11.4,' AK2(Pa)=',0PE11
     *   .4,' ISKIP= ',I2)
 5600 FORMAT (1X,4(1PE11.4,2X))
 5700 FORMAT ('  CrtSts frac    Gc frac     strs frac    dil frac')
C
      END
```

APPENDIX E

FORTRAN LISTING FOR NONLINEAR MICROMECHANICAL MODEL

Sample P14 Input File

```
ct3ms, part intact, vacuole          ! information header
180                                  ! no. pts. in curve
1                                    ! no. distributions
15.5                                 ! avg. particle radius (micron)
0.167                                ! log std. dev.
0.307,0.0                            ! inclusion and void fraction
4609                                 ! sample volume (mm3)
2.1E-4,1.17,0,1,0                    ! Fb,Ym,w-type,m-type,v-type
0.4500E6,30E9                        ! matrix and inclusion shear modulus (Pa)
0.495,0.16                           ! matrix and inclusion Poisson ratio
30E9,34.12E9                         ! matrix and inclusion bulk modulus (Pa)
1.204,0                              ! Gc, pressure
2,1E-3,20                            ! a-type, conv. tol., max. iterations
10,0.45E6,0.0,2                   ! no.steps init.,crit.strs,dbnd angle,avg.fac.
1.554865E6,-0.497499E6,0.321452E6  ! nonlinear co-effs in Young's modulus (Pa)
```

FORTRAN Listing for P14.FOR

```
C==== main program
C      P14.FOR
C====
C     user enters the following information:
C        number of points desired in stress-strain curve after debond
C        number of particle distributions
C        avg radius and std dev of each distribution
C        volume fraction of filler and voids of each distribution
C        sample volume
C        fraction debond, YMULT, w-type, m-type, v-type
C        matrix and filler shear modulus
C        matrix and filler poisson ratio
C        void shear and bulk moduli (if both values zero model as voids,
C          if non-zero, use filler or pseudo-filler values)
C        adhesion energy and applied pressure
C        a-type, convergence tolerance, max. iterations
C        # pts before debonding, critical stress, debond angle, avg.fac.
C        coefficients k0, k1, k2 for fitted matrix modulus
C     information may be entered using keyboard or by input data file.
C     avg.fac. (averaging factor) is number of points to use to
C        determine average energy loss.
C     three options for printing out intermediate results are available:
C        if values for no pts desired in stress-strain curve and number
C           of particle distributions are negative, data files GAUSS,
C           HISTO, DEBUG and STRESS are written.
C        if value for no pts desired in stress-strain curve is negative
C           and value for number of particle distributions is positive,
C           data files HISTO, DEBUG and STRESS are written. Also, all
C           intermediate points where e_i < e_max in the debond section
C           will be written.
C     if input data was entered using data file, the data file STRESS
C     will be renamed to the input data file's name.
C
C     to compile and link: fl pxx.for graphics.lib. the files
C     MSGRAPH.FOR and GRFDEF.FOR should be in the same directory unless
```

```
C     a temporary variable has been set up to point to the location of
C     include files. these files contain graphics routines necessary to
C     plot stress-strain curve on screen.
C
C     first file written by sbr STRWRT has indexing of the various
C     parameters organized as follows:
C     strain  - current critical strain
C     stress  - current true stress calc using previous E and G along
C               with crit.strain
C     Pr_surv - current no. of particles remaining
C     E,G     - moduli at current Pr_surv
C     Vf,Vv   - current filler and void volume fractions
C     dG/dc, dK/dc, dA/dc - current differential quantities
C     the total number of points is NDIST*NPTS+NLTOT+1 where the
C     additional point is for zero strain and stress. the first group of
C     debonded particles begins at ICNT=NLTOT+2.
C
C     calculates particle size histogram with corresponding filler
C     volume fraction. uses Z-decrements for particle size
C     determination
C
C     added routine to output SQRT(r*dE/dc). dc based on total volume
C     instead of Vf+Vm. trapped zero in SQRT calc of crit. strain.
C
C     implementation of Mori-Tanaka solution extended for 3-phase and
C     particle interaction. constituent material properties
C     designated as follows: 1-inclusion,2-void or vacuole,3-matrix.
C     fraction debond (FDBND) for orthotropic properties in loading
C     direction, multiplier for rad. dist. func. (YMULT), w-type
C     designates use inclusion or void properties in
C     calc of Wv matrix (0=void, 1=inclusion), m-type determines type of
C     particle interaction used (0=none,1=inclusion, 2=inclusion and
C     void or vacuole), v-type determines isotropic or orthotropic matl
C     (0=orthotropic,1=isotropic), a-type determines if strain
C     augmentation is used (0=none, 1=yes in initial only, 2=yes in
C     initial and debond sections)
C
```

```
C     implementation of strain-dependent matrix modulus. added COMMON
C     block /PROPC/, rearranged conditional statement calling sbr
C     CALCIO, CALCCV and CMPRPO. included composite strain in the MTPRP
C     and CALCIO variable lists.
C     added sbr NLSTRS to calculate behavior before debonding. have to
C     set critical stress STSFUD to identify when debonding starts.
C     debonding values offset by NLTOT.
C     augmented matrix strain used in initial portion of curve and for
C     calculating matrix properties after each increment in critical
C     strain calc. stored augmented strain in array STRNP1 in /PROPC/.
C
C     added statements in STRWRT to calculate energy released by
C     debonded particles in particle distribution.
C
C     included modification of debonded surface area by debond angle.
C
C     fixed synchronization problem between modulus energy liberation
C     and surface area energy consumption in SBR CRIT.
C
C     last revision: 28 OCT 1996  15H00
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /MATRA/ BETA(2),WI(3,3),WV(3,3),IDENT(3,3)
      COMMON /MATRB/ S(3,3),CA(3,3),CB(3,3),CE(3,3),CF(3,3)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
```

```
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      CHARACTER FILNM*8
C
C==   initialize variables and arrays by BLOCK DATA INIT
C
      CALL INPUT(NDIST,NTOT,VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS,GAMM,
     *   PRESS,FILNM,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,DBANG,ISKIP)
C
      IABORT = 0
      CALL STRSTN(NDIST,NTOT,NPTS,VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *   GAMM,PRESS,DILATO,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,
     *   STRNP,DBANG,ISKIP,IABORT)
C
C==   write out debond only results and debug data
      CALL STRWRT(NDIST,NPTS,VOLSMP,GAMM,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *   PRESS,DILATO,FILNM,NLTOT,STSFUD,IWRT,DBANG,ISKIP,IABORT)
      IF (ABS(IWRT).GE.1) THEN
         CALL STRAUX(NDIST,NPTS,VOLSMP,GAMM,FDBND,YMULT,IKIND,IMORI,
     *   IPOIS,PRESS,DILATO,FILNM,NLTOT,STSFUD,IWRT,DBANG,ISKIP,IABORT)
         CALL DBGWRT(NDIST,NPTS,NLTOT,IABORT)
         CALL DBGRAT(NDIST,NPTS,NLTOT,IABORT)
      ENDIF
C
      CALL CRVPLT(NDIST,NPTS,NLTOT,IWRT,IABORT)
C
      END
C
C
      SUBROUTINE INPUT(NDIST,NTOT,VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *   GAMM,PRESS,FILNM,IWRT,IAUG,STNTOL,ITERMX,NLTOT,STSFUD,DBANG,
     *   ISKIP)
C==== reads in problem input either by file or keyboard. if data entered
C     through a file, user inputs name only, a file extension of DAT is
C     assumed. the first line in the input file is used for a user
```

```
C     heading and is not read in, constituent material properties
C     designated as follows: 1-inclusion,2-void or vacuole,3-matrix
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      CHARACTER ANS*1,FILNM*8
C
      WRITE (6,'(/,A)') ' Read data from file? (Y/N)'
      READ (5,'(A1)') ANS
C
      IF (ANS.EQ.'Y') THEN
         WRITE (6,'(A)') ' File to read? (.INP will be appended)'
         READ (5,'(A8)') FILNM
         OPEN (UNIT=7,FILE=FILNM//'.INP',FORM='FORMATTED',STATUS='OLD')
         READ (7,*)
         READ (7,*) NTOT
         READ (7,*) NDIST
         DO 10 I = 1,ABS(NDIST)
            READ (7,*) RADAVG(I)
            READ (7,*) LOGSTD(I)
            READ (7,*) VLFRFO(I),VLFRVO(I)
   10    CONTINUE
         READ (7,*) VOLSMP
         READ (7,*) FDBND,YMULT,IKIND,IMORI,IPOIS
         READ (7,*) G(3),G(1)
         READ (7,*) POIS(3),POIS(1)
         READ (7,*) G(2),K(2)
         READ (7,*) GAMM,PRESS
         READ (7,*) IAUG,STNTOL,ITERMX
```

```
      READ (7,*) NLTOT,STSFUD,DBANG,ISKIP
      READ (7,*) AK0(1),AK1(1),AK2(1)
      CLOSE (7)
    ELSE
      WRITE (6,'(/,A,I3,A)')
     *    ' no. pts desired in stress-strain curve (<',GSMX,')'
      READ (5,*) NTOT
      WRITE (6,'(A,I1,A)') ' no. of particle distributions (<=',
     *    NTDIS,')'
      READ (5,*) NDIST
      DO 20 I = 1,ABS(NDIST)
        WRITE (6,'(A,I1,A)') ' for distribution no. ',I,
     *      ' mean radius (micron)'
        READ (5,*) RADAVG(I)
        WRITE (6,'(A)') ' log normal radius std dev'
        READ (5,*) LOGSTD(I)
        WRITE (6,'(A)') ' initial volume fraction filler and void'
        READ (5,*) VLFRF0(I),VLFRV0(I)
20    CONTINUE
      WRITE (6,'(A)') ' sample volume (mm3)'
      READ (5,*) VOLSMP
      WRITE (6,'(A)') ' dbnd frac,rad dist mult,w-type,m-type,v-type'
      READ (5,*) FDBND,YMULT,IKIND,IMORI,IPOIS
      WRITE (6,'(A)') ' matrix and filler shear modulus (Pa)'
      READ (5,*) G(3),G(1)
      WRITE (6,'(A)') ' matrix and filler Poisson ratio'
      READ (5,*) POIS(3),POIS(1)
      WRITE (6,'(A)') ' void shear and bulk modulus (Pa)'
      READ (5,*) G(2),K(2)
      WRITE (6,'(A)') ' Gc (J/m2) and applied pressure (Pa)'
      READ (5,*) GAMM,PRESS
      WRITE (6,'(A)') ' a-type, strain tolerance, max. iterations'
      READ (5,*) IAUG,STNTOL,ITERMX
      WRITE (6,'(A)') ' # pts before debonding, crit strs(Pa), debond
     * ang(deg), default avg.fac.'
      READ (5,*) NLTOT,STSFUD,DBANG,ISKIP
      WRITE (6,'(A)') ' matrix tensile modulus coeffs k0,k1,k2 (Pa)'
```

```
         READ (5,*) AK0(1),AK1(1),AK2(1)
         FILNM = 'DEFAULT'
      ENDIF
C
C==   set write file flag, 0=STRWRT, 1=STRWRT,DBGWRT,HSTWRT, 2=all
      IWRT = 0
      IF (NTOT.LT.0.AND.NDIST.LT.0) IWRT = 2
      IF (NTOT.LT.0.AND.NDIST.GT.0) IWRT = 1
      NDIST = ABS(NDIST)
      NTOT = ABS(NTOT)
      IF (IAUG.EQ.0) NLTOT=0
C
      RETURN
      END
      SUBROUTINE STRSTN(NDIST,NTOT,NPTS,VOLSMP,FDBND,YMULT,IKIND,IMORI,
     *   IPOIS,GAMM,PRESS,DILAT0,IWRT,IAUG,STNTOL,ITERMX,NLTOT,
     *   STSFUD,STRNP,DBANG,ISKIP,IABORT)
C==== main subroutine which organizes particle size distribution,
C     composite property, critical strain and true stress and dilation
C     calculation modules.
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AK0(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
C
C==   initialize abort flag
      IABORT = 0
```

```
C
C==   create gaussian distribution of particle size
      WRITE (6,'(/,A)') ' Generating particle distribution'
      CALL GAUSS(NDIST,NTOT,NPTS,IABORT)
C==   write out gaussian cumulative data
      IF (ABS(IWRT).GE.2) CALL GAUWRT(NDIST,NPTS,IABORT)
C
C==   find size and number of particles to debond
      WRITE (6,'(/,A)') ' Finding particle size and number'
      CALL PARTSZ(NDIST,NPTS,VOLSMP,IABORT)
C==   write out particle size and number histogram
      IF (ABS(IWRT).GE.1) CALL HSTWRT(NDIST,NPTS,IABORT)
C
      WRITE (6,'(/,A)') ' Sorting particle distributions'
      CALL SORTER(NDIST,NPTS,IABORT)
      WRITE (6,'(A)') ' Calculating vol fractions and dA/dc'
      CALL VOLFRC(NDIST,NPTS,VOLSMP,DBANG,IABORT)
C
      WRITE (6,'(/,A)') ' Generating true stress-strain curve'
C==   calculate initial composite properties
      ICNT = 1
      CONCI = NETVF(ICNT)
      CONCV = NETVV(ICNT)
      STRNO=0
      CALL MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,IPOIS,STRNO,
     *   IAUG,IABORT)
      IF (IABORT.EQ.0) WRITE (6,'(A,1X,I3,A,I3,A)')
     *   ' Calculating point: ',ICNT,'/',NDIST*NPTS+NLTOT+1,' max'
      CALL NLSTRS(NDIST,NPTS,FDBND,YMULT,IKIND,IMORI,IPOIS,IAUG,STNTOL,
     *   ITERMX,NLTOT,STSFUD,STRNP,DILATO,IABORT)
C
C==   main routine for debonding and true stress-strain calculation.
C     offset pointer ICNT by 1 to make room for undebonded state.
C     STRNP in NLSTRS is matrix strain at end of initial portion.
C     matrix strain revised if e_i > e_max, if it is not STRNP
C     maintained at current e_max. IPDIST used as a flag to show when
C     STRNP has been updated.
```

```
C
      CRTMAX = CRTSTN(NLTOT+1)
      DO 50 ICNT = 2+NLTOT,NDIST*NPTS+NLTOT+1
         CONCI = NETVF(ICNT)
         CONCV = NETVV(ICNT)
         CALL MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,
     *      IPOIS,STRNP,IAUG,IABORT)
         IF (IABORT.EQ.0) WRITE (6,'(A,1X,I3,A,I3,A)')
     *      ' Calculating point: ',ICNT,'/',NDIST*NPTS+NLTOT+1,' max'
         CALL CRIT(ICNT,NLTOT,IAUG,VOLSMP,GAMM,PRESS,CRTMAX,STRNP,
     *      ISKIP,IABORT)
         STRNP1(ICNT) = STRNP
         CALL CALVAL(ICNT,PRESS,DILATO,IABORT)
   50 CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE GAUSS(NDIST,NTOT,NPTS,IABORT)
C==== Program calculates the cumulative area underneath the
C     gaussian curve between the limits +/- (IEND/FACT) in increments
C     of IDELT/FACT. NTOT is used to calculate an appropriate IDELT.
C     since IDELT is rounded down, the exact number of points may be
C     greater. this is reflected in NPTS.
C     Particle radii converted from microns to millimeters.
C     An IEND of 3301 gives a cumulative distribution which starts
C     at 0.0005 and ends at 0.9995. This avoids having extremely large
C     particles when the log standard deviation is large.
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      PARAMETER (ISTART = 2,IEND = 3301,FACT = 1000,BEGNPT = 0)
      EXTERNAL FUNC
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
```

```
     *   VLFRVO(NTDIS)
C
      IMAX = GSMX/2
      IDELT = 2*INT((IEND-ISTART)/NTOT)
      NPTS = 2*(INT((IEND-ISTART)/IDELT)+1)
      IF (NPTS.GT.IMAX) THEN
         WRITE (6,'(A)') ' Too many points: SBR GAUSS.'
         WRITE (6,'(A,I4,A)') ' Over max dim by ',NPTS-GSMX,' points.'
         IABORT = 1
         RETURN
      ELSE
      ENDIF
C
      DO 20 J = 1,NDIST
         IPTS = 0
         DO 10 I = ISTART,IEND,IDELT
            IPTS = IPTS+1
            ENDPT = (FLOAT(I)-1)/FACT
            Z(NPTS/2+IPTS) = ENDPT
            Z(NPTS/2-IPTS+1) = -ENDPT
C==   calculate upper portion of probability curve
            RADTMP = 10**(ALOG10(RADAVG(J))+ENDPT*LOGSTD(J))
            RADIUS(J,NPTS/2+IPTS) = RADTMP/1000
            CALL QSIMP(FUNC,BEGNPT,ENDPT,SURF)
            PROB(J,NPTS/2+IPTS) = 0.5+SURF
C==   calculate lower portion of probability curve
            RADTMP = 10**(ALOG10(RADAVG(J))-ENDPT*LOGSTD(J))
            RADIUS(J,NPTS/2-IPTS+1) = RADTMP/1000
            PROB(J,NPTS/2-IPTS+1) = 0.5-SURF
   10    CONTINUE
   20 CONTINUE
C
      RETURN
      END
C
      FUNCTION FUNC(X)
C==== function used for gaussian curve. called from sbr GAUSS, sbr
```

```
C     QSIMP and sbr TRAPZD.
      PI = 3.141592654
      FUNC = (1.0/SQRT(2.0*PI))*EXP(-X**2/2.0)
      RETURN
      END
C
C
```

```
      SUBROUTINE PARTSZ(NDIST,NPTS,VOLSMP,IABORT)
C==== sbr finds the total particle volume on a per particle basis.
C     from this the number of particles present in the composite is
C     calculated knowing the initial volume the particles occupy.
C     the incremental probability of the largest particles is
C     calculated using a fraction (PFRAC) of the previous probability
C     increment so that there is a smooth transition from largest to
C     smaller particle sizes in terms of number.
C
C     set NPTMX = GSMX*NTDIS
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      PARAMETER (PI = 3.1415927,PFRAC = 0.75)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
C
      IF (IABORT.EQ.1) RETURN
C
C==   find total number of particles in given filler volume
      DO 20 IDIST = 1,NDIST
         VOLTOT = 0
C     find total volume on a per particle basis
         DO 10 IPTS = 1,NPTS
            IF (IPTS.EQ.NPTS) THEN
               VOLPAR(IDIST,IPTS) = PFRAC*(PROB(IDIST,IPTS)-
     *            PROB(IDIST,IPTS-1))*(4.0/3.0)*PI*RADIUS(IDIST,IPTS)**3
            ELSE
               VOLPAR(IDIST,IPTS) = (PROB(IDIST,IPTS+1)-
     *            PROB(IDIST,IPTS))*(4.0/3.0)*PI*RADIUS(IDIST,IPTS)**3
            ENDIF
            VOLTOT = VOLTOT+VOLPAR(IDIST,IPTS)
   10    CONTINUE
```

```
C     find total number of particles
         NPARTL(IDIST) = VLFRFO(IDIST)*VOLSMP/VOLTOT
   20 CONTINUE
C
C     calculate volume taken up by particles of radius r
      DO 40 IDIST = 1,NDIST
         DO 30 IPTS = 1,NPTS
            IF (IPTS.EQ.NPTS) THEN
               NUMPAR(IDIST,IPTS) = NPARTL(IDIST)*PFRAC*(
     *            PROB(IDIST,IPTS)-PROB(IDIST,IPTS-1))
               IF (NUMPAR(IDIST,IPTS).LT.1.0) IFLAG = 1
               VOLPAR(IDIST,IPTS) = NUMPAR(IDIST,IPTS)*(4.0/3.0)*PI*
     *            RADIUS(IDIST,IPTS)**3
            ELSE
               NUMPAR(IDIST,IPTS) = NPARTL(IDIST)*(PROB(IDIST,IPTS+1)-
     *            PROB(IDIST,IPTS))
               IF (NUMPAR(IDIST,IPTS).LT.1.0) IFLAG = 1
               VOLPAR(IDIST,IPTS) = NUMPAR(IDIST,IPTS)*(4.0/3.0)*PI*
     *            RADIUS(IDIST,IPTS)**3
            ENDIF
C
            IF (IFLAG.EQ.1) THEN
               WRITE (6,5000) IDIST,IPTS,RADIUS(IDIST,IPTS),
     *            NUMPAR(IDIST,IPTS)
               IFLAG = 0
            ELSE
            ENDIF
C
   30    CONTINUE
   40 CONTINUE
C
C     RETURN
 5000 FORMAT (' Error SBR PARTSZ: IDIST=',I1,' IPTS=',I3,' RAD=',E11.6,
     *   ' NUMPAR=',E11.6)
      END
C
C
```

```
      SUBROUTINE SORTER(NDIST,NPTS,IABORT)
C==== loads radius, number of particles and total volume of particles
C     of radius r from each distribution in a master array to sort.
C     after sorting radii in ascending order, arrays are flipped
C     according to radius to give descending order.
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
C
      IF (IABORT.EQ.1) RETURN
C
C*    load master arrays
      DO 20 I = 1,NDIST
         DO 10 J = 1,NPTS
            SORRAD((I-1)*NPTS+J) = RADIUS(I,J)
            SORPAR((I-1)*NPTS+J) = NUMPAR(I,J)
            SORVLP((I-1)*NPTS+J) = VOLPAR(I,J)
            IPDIST((I-1)*NPTS+J) = I
   10    CONTINUE
   20 CONTINUE
C
C*    sort master arrays in ascending order
      NTOT = NDIST*NPTS
      CALL SORT3(NTOT,SORRAD,SORPAR,SORVLP,IPDIST)
C
C*    sort master arrays in descending order
      DO 30 I = 1,NDIST*NPTS/2
         ATMP = SORRAD(I)
```

```
            SORRAD(I) = SORRAD(NDIST*NPTS-I+1)
            SORRAD(NDIST*NPTS-I+1) = ATMP
            BTMP = SORPAR(I)
            SORPAR(I) = SORPAR(NDIST*NPTS-I+1)
            SORPAR(NDIST*NPTS-I+1) = BTMP
            CTMP = SORVLP(I)
            SORVLP(I) = SORVLP(NDIST*NPTS-I+1)
            SORVLP(NDIST*NPTS-I+1) = CTMP
            ITMP = IPDIST(I)
            IPDIST(I) = IPDIST(NDIST*NPTS-I+1)
            IPDIST(NDIST*NPTS-I+1) = ITMP
   30 CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE VOLFRC(NDIST,NPTS,VOLSMP,DBANG,IABORT)
C==== calculates dA/dc, net Vf, net Vv and probility of survival for
C     given particle radius. Note: net Vf is based on total sample vol.
C     Prob of surv is based on numbers of particles.
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      PARAMETER (PI = 3.1415927)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
C
      IF (IABORT.EQ.1) RETURN
```

```
C
C==   calculate total volume fraction filler and void
      NETVF(1) = 0
      NETVV(1) = 0
      DO 10 I = 1,NDIST
         NETVF(1) = NETVF(1)+VLFRFO(I)
         NETVV(1) = NETVV(1)+VLFRVO(I)
   10 CONTINUE
      PRBSRV(1) = 1.0
C
C==   calculate net Vf and Vv, dA/dc and Prob surv. array index offset
C     by 1 to leave room for initial undebonded state and leave room for
C     initial portion of stress-strain curve
C
      SRVNUM = 0
C
C==   find total number of particles
      DO 20 ICNT = 1,NDIST
         TLNUMP = TLNUMP+NPARTL(ICNT)
   20 CONTINUE
C
      DO 30 JCNT = 2,NDIST*NPTS+1
         VLFTOT = VLFTOT-SORVLP(JCNT-1)
         NETVF(JCNT) = NETVF(JCNT-1)-SORVLP(JCNT-1)/VOLSMP
         NETVV(JCNT) = NETVV(JCNT-1)+SORVLP(JCNT-1)/VOLSMP
         SRVNUM = SRVNUM+SORPAR(JCNT-1)
         PRBSRV(JCNT) = (TLNUMP-SRVNUM)/TLNUMP
         DADC(JCNT) = -6.0*(1-SIN(DBANG*3.14159/180))*VOLSMP/
     *       SORRAD(JCNT-1)
   30 CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *   STNTMP,IAUG,IABORT)
```

```
C==== program for calculating composite modulus based on Mori-Tanaka.
C     FDBND=fraction debond for orthotropic properties in loading
C     direction, IKIND=use inclusion or void properties in calc of
C     Wv matrix, IMORI=type of particle interaction used 0=none,
C     1=inclusion, 2=inclusion and void or vacuole,IPOIS=type of
C     debond properties 0=orthotropic,1=isotropic. recalculates
C     matrix modulus each time routine is called.
C
      REAL IDENT,K,KCMP,MAG
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      DIMENSION CAVG(3,3)
C
      IF (IABORT.EQ.1) RETURN
C
C     changed this block so that any variables that depend on matrix
C     modulus calculated each time new strain is available
      CALL CALCIO(STNTMP,IAUG,IABORT)
      IF (ICNT.EQ.1) CALL CALCCV(FDBND,IPOIS,IABORT)
      CALL CMPRPO(IKIND,IMORI,IABORT)
C
      CALL CMPRP(CONCI,CONCV,YMULT,CAVG,IABORT)
      C11(ICNT) = CAVG(1,1)
      C12(ICNT) = CAVG(1,2)
      C21(ICNT) = CAVG(2,1)
      C22(ICNT) = CAVG(2,2)
      C23(ICNT) = CAVG(2,3)
      ECMP(ICNT) = C11(ICNT)-2.0*C12(ICNT)*C21(ICNT)/(C22(ICNT)+
     *   C23(ICNT))
      POISC(ICNT) = C21(ICNT)/(C22(ICNT)+C23(ICNT))
C
      RETURN
      END
C
C
      SUBROUTINE CRIT(ICNT,NLTOT,IAUG,VOLSMP,GAMM,PRESS,CRTMAX,
```

```
     *   STRNP,ISKIP,IABORT)
C==== calculates current critical strain based on difference between
C     current and previous properties because the energy balance
C     requires the input work to equal the energy released by surface
C     creation and the internal energy stored after debonding has taken
C     place.
C     surface area energy consumption is averaged over IPDIST=9 points
C     and used with energy liberated due to reinforcement loss in
C     increment ICNT. the energy in ICNT can be considered a rough
C     estimate of the avg. loss between IPDIST=9 points.
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,KTMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      PARAMETER (TOL = 1E-18)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
C
      IF (IABORT.EQ.1) RETURN
C
      CONV = 1.0E+3
C
      IFACT = 0
      DADCTL = DADC(ICNT)
      IF (IPDIST(ICNT-1).NE.9.AND.ICNT.GT.NLTOT+2) THEN
         IFACT = 1
         DO 10 I = ICNT-1,NLTOT+2,-1
            IF (IPDIST(I).EQ.9) GOTO 20
            DADCTL = DADCTL + DADC(I)
```

```
            IFACT = IFACT+1
   10     CONTINUE
   20  CONTINUE
       ENDIF
C
       DC = NETVF(ICNT)-NETVF(ICNT-1)
       IF (ABS(DC).LT.TOL) DC = -TOL
C
       TC12 = C12(ICNT)
       TC21 = C21(ICNT)
       TC22 = C22(ICNT)
       TC23 = C23(ICNT)
       DC11 = (C11(ICNT)-C11(ICNT-1))/DC
       DC12 = (C12(ICNT)-C12(ICNT-1))/DC
       DC21 = (C21(ICNT)-C21(ICNT-1))/DC
       DC22 = (C22(ICNT)-C22(ICNT-1))/DC
       DC23 = (C23(ICNT)-C23(ICNT-1))/DC
C
       AQUAD = -DC11+2.0*((TC22+TC23)*(TC21*DC12+TC12*DC21)-(TC12*TC21*(
      *   DC22+DC23)))/(TC22+TC23)**2
       CQUAD = CONV*2*GAMM*DADCTL/VOLSMP
C
       IF (IFACT.EQ.0) IFACT=ISKIP
       CQUAD = CQUAD/REAL(IFACT)
C
       IF (AQUAD.GE.0) THEN
          WRITE (6,'(A)') ' SBR CRIT: square root term is negative.'
          CRTSTN(ICNT) = CRTSTN(ICNT-1)
       ELSE
          CRTSTN(ICNT) = SQRT(CQUAD/AQUAD)
       ENDIF
C
       IF (CRTSTN(ICNT).GT.CRTMAX) THEN
          STRNP = (ECMP(ICNT)/E(3))*CRTSTN(ICNT)
          CRTMAX = CRTSTN(ICNT)
          IPDIST(ICNT) = 9
       ENDIF
```

```
C
      RETURN
      END
C
C
      SUBROUTINE CALVAL(ICNT,PRESS,DILATO,IABORT)
C==== calculates true stress and dilatation at critical strain
C     properties used are those before debonding takes place
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,KTMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
C
      IF (IABORT.EQ.1) RETURN
C
      TC21 = C21(ICNT-1)
      TC22 = C22(ICNT-1)
      TC23 = C23(ICNT-1)
      ETMP = ECMP(ICNT-1)
      IF (ICNT.EQ.2) DILATO = PRESS*0
C
      STRESS(ICNT) = ETMP*CRTSTN(ICNT)
C==   change stress values to MPa
      STRESS(ICNT) = STRESS(ICNT)/1.0E6
      DILAT(ICNT) = (1-(2.0*TC21/(TC22+TC23)))*CRTSTN(ICNT)-DILATO
C
      RETURN
      END
C
C
      SUBROUTINE NLSTRS(NDIST,NPTS,FDBND,YMULT,IKIND,IMORI,IPOIS,IAUG,
```

```
     *    STNTOL,ITERMX,NLTOT,STSFUD,STRNP,DILATO,IABORT)
C==== offsets results to make room for pre-debonding results. uses
C     STSFUD as max. stress and sub-divides interval into NLTOT sub-
C     steps. iterates to find equilibrium strain.
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      PARAMETER (PRMX = 0.95)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *    NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *    C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *    PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *    IPDIST(NPTMX)
C
      IF (IABORT.EQ.1.OR.IAUG.EQ.0) RETURN
C
C     shift values which are needed later by sbr STRSTN
      DO 10 I=NDIST*NPTS+1,2,-1
         NETVF(I+NLTOT) = NETVF(I)
         NETVV(I+NLTOT) = NETVV(I)
         DADC(I+NLTOT) = DADC(I)
         PRBSRV(I+NLTOT) = PRBSRV(I)
   10 CONTINUE
C
C     fill in the values that do not change in this stage
      DO 20 I=2,NLTOT+1
         PRBSRV(I) = 1
         NETVF(I) = NETVF(1)
         NETVV(I) = NETVV(1)
         IPDIST(I) = 0
   20 CONTINUE
```

```
C
C     calculate stress-strain behavior
      CONCI = NETVF(1)
      CONCV = NETVV(1)
      DO 70 ICNT = 2,NLTOT+1
         STRESS(ICNT) = REAL(ICNT-1)*STSFUD/REAL(NLTOT)
C
         STRNO = 0.0
         IF (ICNT.GT.2) STRNO = CRTSTN(ICNT-1)
         DO 50 ITER = 1,ITERMX
            CALL MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,IPOIS,
     *         STRNO,IAUG,IABORT)
C
            IF (ABS(IAUG).GE.1) THEN
               STRNPO = (ECMP(ICNT)/E(3))*STRNO
               DO 30 JTER = 1,ITERMX
                  CALL MTPRP(CONCI,CONCV,ICNT,FDBND,YMULT,IKIND,IMORI,
     *               IPOIS,STRNPO,IAUG,IABORT)
                  STRNP1(ICNT) = (ECMP(ICNT)/E(3))*STRNO
                  ERR = 1
                  IF (STRNP1(ICNT).NE.0) ERR = ABS((STRNP1(ICNT)
     *               -STRNPO)/STRNP1(ICNT))
C                 WRITE (6,'(A,2X,I2,2X,I2,2X,I2,3E12.4)')
C    *               ' ICNT-ITER-JTER-P1-PO-ERR',ICNT,ITER,JTER,STRNP1,
C    *               STRNPO,ERR
               STRNP = STRNP1(ICNT)
               IF (ERR.LE.STNTOL) GOTO 40
               STRNPO = STRNP1(ICNT)
   30       CONTINUE
            IF (ITER.GE.ITERMX.OR.JTER.GE.ITERMX) WRITE (6,'(A,I3)')
     *         ' SBR NLSTRS: iteration max. reached. ICNT=',ICNT
   40       CONTINUE
         ELSE
         ENDIF
C
            STRN1 = STRESS(ICNT)/ECMP(ICNT)
            ERR = ABS((STRN1-STRNO)/STRN1)
```

```
C           WRITE (6,'(A,2X,I2,2X,I2,4E12.4)')
C     *        ' ICNT-ITER-STRS-ECMP-R-ERR',
C     *        ICNT,ITER,STRESS(ICNT),ECMP(ICNT),ECMP(ICNT)/E(3),ERR
            IF (ERR.LE.STNTOL) GOTO 60
            STRN0 = STRN1
   50     CONTINUE
   60     CONTINUE
C          change true stress values to MPa
          STRESS(ICNT) = STRESS(ICNT)/1E6
          CRTSTN(ICNT) = STRN1
          TC21 = C21(ICNT)
          TC22 = C22(ICNT)
          TC23 = C23(ICNT)
          DILAT(ICNT) = (1-(2.0*TC21/(TC22+TC23)))*STRN1+DILAT0
C
          IF (IABORT.EQ.0) WRITE (6,'(A,1X,I3,A,I3,A)')
     *       ' Calculating point: ',ICNT,'/',NDIST*NPTS+NLTOT+1,' max'
C
   70 CONTINUE
C
       RETURN
       END
C
       BLOCK DATA INIT
C==== initialize all variables and arrays used in program
C      check NPTMX if NTDIS or GSMX are changed.
C         NPTMX = NTDIS*GSMX
C
       REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
       REAL IDENT,K,KCMP,MAG
       INTEGER GSMX,NPTMX
       PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
       COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
       COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
      *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
       COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRF0(NTDIS),
      *   VLFRV0(NTDIS)
```

```
      COMMON /MATRA/ BETA(2),WI(3,3),WV(3,3),IDENT(3,3)
      COMMON /MATRB/ S(3,3),CA(3,3),CB(3,3),CE(3,3),CF(3,3)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AK0(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
C
      DATA Z /GSMX*0/ RADIUS /NPTMX*0/ PROB /NPTMX*0/
      DATA NUMPAR /NPTMX*0/ VOLPAR /NPTMX*0/ NETVF /NPTMX*0/ NETVV /
     *   NPTMX*0/ DADC /NPTMX*0/ NPARTL /NTDIS*0/
      DATA RADAVG /NTDIS*0/ LOGSTD /NTDIS*0/ VLFRF0 /NTDIS*0/ VLFRV0 /
     *   NTDIS*0/
      DATA BETA /2*0/ WI /9*0/ WV /9*0/ IDENT /1,0,0,0,1,0,0,0,1/
      DATA S /9*0/ CA /9*0/ CB /9*0/ CE /9*0/ CF /9*0/
      DATA K /3*0/ G /3*0/ E /3*0/ POIS /3*0/ CI /9*0/ CV /9*0/ CO /9*0/
      DATA C11 /NPTMX*0/ C12 /NPTMX*0/ C21 /NPTMX*0/ C22 /NPTMX*0/ C23 /
     *   NPTMX*0/ ECMP /NPTMX*0/ POISC /NPTMX*0/
      DATA AK0 /1*0/ AK1 /1*0/ AK2 /1*0/ STRNP1 /NPTMX*0/
      DATA CRTSTN /NPTMX*0/ STRESS /NPTMX*0/ DILAT /NPTMX*0/ PRBSRV /
     *   NPTMX*0/ SORRAD /NPTMX*0/ SORPAR /NPTMX*0/ SORVLP /NPTMX*0/
     *   IPDIST /NPTMX*0/
C
      END
```

```
      SUBROUTINE QSIMP(FUNC,A,B,S)
C==== used for integration of gaussian curve in sbr GAUSS. obtained
C     from Numerical Recipes, W.H. Press, Cambridge, 1988.
C
      EXTERNAL FUNC
      PARAMETER (EPS = 1.E-6,JMAX = 20)
      OST = -1.E30
      OS = -1.E30
      DO 10 J = 1,JMAX
         CALL TRAPZD(FUNC,A,B,ST,J)
         S = (4.*ST-OST)/3.
         IF (ABS(S-OS).LT.EPS*ABS(OS)) RETURN
         OS = S
         OST = ST
   10 CONTINUE
      PAUSE 'Too many steps: SBR QSIMP'
      END
C
C
      SUBROUTINE TRAPZD(FUNC,A,B,S,N)
C==== used for integration of gaussian curve in sbr QSIMP which is
C     called from sbr GAUSS. obtained from Numerical Recipes, W.H.
C     Press, Cambridge, 1988.
C
      EXTERNAL FUNC
      IF (N.EQ.1) THEN
         S = 0.5*(B-A)*(FUNC(A)+FUNC(B))
         IT = 1
      ELSE
         TNM = IT
         DEL = (B-A)/TNM
         X = A+0.5*DEL
         SUM = 0.
         DO 10 J = 1,IT
            SUM = SUM+FUNC(X)
            X = X+DEL
   10    CONTINUE
```

```
            S = 0.5*(S+(B-A)*SUM/TNM)
            IT = 2*IT
         ENDIF
         RETURN
         END
C
C
         SUBROUTINE SORT3(N,RA,RB,RC,IRD)
C==== sorting routine from Numerical Recipes, W.H. Press, Cambridge,
C     1988. sorts in ascending order array RA and moves elements in
C     arrays RB, RC and IRD at the same time.
C
         DIMENSION RA(N),RB(N),RC(N),IRD(N)
         L = N/2+1
         IR = N
      10 CONTINUE
         IF (L.GT.1) THEN
            L = L-1
            RRA = RA(L)
            RRB = RB(L)
            RRC = RC(L)
            IRRD = IRD(L)
         ELSE
            RRA = RA(IR)
            RRB = RB(IR)
            RRC = RC(IR)
            IRRD = IRD(IR)
            RA(IR) = RA(1)
            RB(IR) = RB(1)
            RC(IR) = RC(1)
            IRD(IR) = IRD(1)
            IR = IR-1
            IF (IR.EQ.1) THEN
               RA(1) = RRA
               RB(1) = RRB
               RC(1) = RRC
               IRD(1) = IRRD
```

```
            RETURN
         ENDIF
      ENDIF
      I = L
      J = L+L
   20 IF (J.LE.IR) THEN
         IF (J.LT.IR) THEN
            IF (RA(J).LT.RA(J+1)) J = J+1
         ENDIF
         IF (RRA.LT.RA(J)) THEN
            RA(I) = RA(J)
            RB(I) = RB(J)
            RC(I) = RC(J)
            IRD(I) = IRD(J)
            I = J
            J = J+J
         ELSE
            J = IR+1
         ENDIF
         GOTO 20
      ENDIF
      RA(I) = RRA
      RB(I) = RRB
      RC(I) = RRC
      IRD(I) = IRRD
      GOTO 10
      END
C
C
      SUBROUTINE CALCIO(STNTMP,IAUG,IABORT)
C==== calculate the property matrix for inclusion and matrix,
C     isotropic relations. have assumed that matrix tensile
C     modulus can be fitted to a 3rd order polynomial.
C
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
```

```
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPC/ AK0(1),AK1(1),AK2(1),STRNP1(NPTMX)
C
      IF (IABORT.EQ.1) RETURN
C
C     make matrix shear modulus dependent on strain if NL analysis
C     desired (IAUG>0). chose this form to make compatible with existing
C     program and allow the tensile modulus curve-fitted parameters to
C     be entered.
      IF (IAUG.GT.0) G(3) = (AK0(1)+AK1(1)*STNTMP+AK2(1)*STNTMP**2)/
     *   (2.0*(1+POIS(3)))
C
      K(1) = (2.0*G(1)*(1+POIS(1)))/(3.0*(1.0-2.0*POIS(1)))
      E(1) = G(1)*(2.0*(1+POIS(1)))
      K(3) = (2.0*G(3)*(1+POIS(3)))/(3.0*(1.0-2.0*POIS(3)))
      E(3) = G(3)*(2.0*(1+POIS(3)))
      C1 = K(1)+(4.0/3.0)*G(1)
      C2 = K(1)-(2.0/3.0)*G(1)
      C3 = K(3)+(4.0/3.0)*G(3)
      C4 = K(3)-(2.0/3.0)*G(3)
      DO 20 I = 1,3
         DO 10 J = 1,3
            CI(I,J) = C2
            CO(I,J) = C4
            IF (I.EQ.J) CI(I,J) = C1
            IF (I.EQ.J) CO(I,J) = C3
   10    CONTINUE
   20 CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE CALCCV(FDBND,IPOIS,IABORT)
C==== calculate the property matrix for debonded particle,
C     orthotropic relations, FDBND is debond fraction for vacuole
C     IPOIS determines whether orthotropic or isotropic
```

```
C
      REAL IDENT,K,KCMP,MAG
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
C
      IF (IABORT.EQ.1) RETURN
C
      IF (K(2).NE.0.AND.G(2).NE.0) THEN
         POIS(2) = (3.0*K(2)-2.0*G(2))/(2.0*(3.0*K(2)+G(2)))
         E(2) = 9.0*K(2)*G(2)/(3.0*K(2)+G(2))
      ELSE
         E(2) = 0.0
         POIS(2) = 0.0
      ENDIF
C
      PCON = REAL(IPOIS)
      DETM = 1-POIS(2)**2-PCON*2*(POIS(2)**2+POIS(2)**3)
      CV(1,1) = (FDBND*E(2)*(1-POIS(2)**2))/DETM
      CV(1,2) = (FDBND*E(2)*(POIS(2)+POIS(2)**2))/DETM
      CV(1,3) = CV(1,2)
      CV(2,1) = (E(2)*PCON*POIS(2)*(1+POIS(2)))/DETM
      CV(2,2) = (E(2)*(1-PCON*POIS(2)**2))/DETM
      CV(2,3) = (E(2)*(POIS(2)+PCON*POIS(2)**2))/DETM
      CV(3,1) = CV(2,1)
      CV(3,2) = CV(2,3)
      CV(3,3) = CV(2,2)
C
      RETURN
      END
C
C
      SUBROUTINE CMPRPO(IKIND,IMORI,IABORT)
C==== calculate constants in composite equation
C
      REAL IDENT,K,KCMP,MAG
      COMMON /MATRB/ S(3,3),CA(3,3),CB(3,3),CE(3,3),CF(3,3)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      DIMENSION CTEMPA(3,3),CTEMPB(3,3)
```

```
C
      IF (IABORT.EQ.1) RETURN
C
      CALL CALCW(IKIND,IMORI,IABORT)
      CALL CALCS(IABORT)
      CALL SUB(CTEMPA,CI,CO)
      CALL INVERT(CTEMPB,CTEMPA,IABORT)
      CALL MULT(CA,CTEMPB,CO)
      CALL SUB(CTEMPA,CV,CO)
      CALL INVERT(CTEMPB,CTEMPA,IABORT)
      CALL MULT(CB,CTEMPB,CO)
      CALL ADD(CE,S,CB)
      CALL ADD(CF,S,CA)
C
      RETURN
      END
C
      SUBROUTINE CMPRP(CONCI,CONCV,YMULT,CAVG,IABORT)
C==   calculate composite properties, ITYPE identifies inclusion
C     or void
      REAL IDENT,K,KCMP,MAG
      COMMON /MATRA/ BETA(2),WI(3,3),WV(3,3),IDENT(3,3)
      COMMON /MATRB/ S(3,3),CA(3,3),CB(3,3),CE(3,3),CF(3,3)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      DIMENSION CC(3,3),CD(3,3),CG(3,3),CH(3,3)
      DIMENSION CTEMPA(3,3),CTEMPB(3,3),CTEMPC(3,3),CAVG(3,3)
C
C==   calculate phase-dependent components of composite equation
C     calculate first half
C     calculate phase-i components
      ITYPE = 1
      CALL GAMMA(CG,CONCI,ITYPE,YMULT,IABORT)
      CALL SUB(CTEMPA,IDENT,S)
      CALL SUB(CTEMPB,CTEMPA,CG)
      DO 20 I = 1,3
         DO 10 J = 1,3
            CC(I,J) = CONCI*CTEMPB(I,J)
```

```
   10    CONTINUE
   20 CONTINUE
C     calculate phase-v components
      ITYPE = 2
      CALL GAMMA(CH,CONCV,ITYPE,YMULT,IABORT)
      CALL SUB(CTEMPA,IDENT,S)
      CALL SUB(CTEMPB,CTEMPA,CH)
      DO 40 I = 1,3
         DO 30 J = 1,3
            CD(I,J) = CONCV*CTEMPB(I,J)
   30    CONTINUE
   40 CONTINUE
      CALL INVERT(CTEMPA,CE,IABORT)
      CALL MULT(CTEMPB,CD,CTEMPA)
      CALL MULT(CTEMPA,CTEMPB,CF)
C     combine phase-i and phase-v components
      CALL ADD(CTEMPB,CTEMPA,CA)
      CALL ADD(CTEMPA,CTEMPB,S)
      CALL ADD(CTEMPB,CTEMPA,CC)
      CALL INVERT(CTEMPA,CTEMPB,IABORT)
      CALL MULT(CTEMPB,CG,CTEMPA)
      DO 60 I = 1,3
         DO 50 J = 1,3
            CTEMPC(I,J) = CONCI*CTEMPB(I,J)
   50    CONTINUE
   60 CONTINUE
C     calculate second half
      CALL INVERT(CTEMPA,CF,IABORT)
      CALL MULT(CTEMPB,CC,CTEMPA)
      CALL MULT(CTEMPA,CTEMPB,CE)
C     combine phase-i and phase-v components
      CALL ADD(CTEMPB,CTEMPA,CB)
      CALL ADD(CTEMPA,CTEMPB,S)
      CALL ADD(CTEMPB,CTEMPA,CD)
      CALL INVERT(CTEMPA,CTEMPB,IABORT)
      CALL MULT(CTEMPB,CH,CTEMPA)
      DO 80 I = 1,3
```

```
         DO 70 J = 1,3
            CTEMPA(I,J) = CONCV*CTEMPB(I,J)
   70    CONTINUE
   80 CONTINUE
C=    combine all components
      CALL ADD(CTEMPB,CTEMPC,CTEMPA)
      CALL ADD(CTEMPA,CTEMPB,IDENT)
      CALL MULT(CAVG,CO,CTEMPA)
C
      RETURN
      END
C
C
      SUBROUTINE CALCW(IKIND,IMORI,IABORT)
C==== calculate correction matrices WI and WV and BETA for
C     use in sbr GAMMA, IKIND determines inclusion or void for vacuole
C     IMORI determines if correction matrix used, 0=none,1=inclusion
C     2=inclusion and void
C
      REAL IDENT,K,KCMP,MAG
      REAL KTEMP,KMAT
      COMMON /MATRA/ BETA(2),WI(3,3),WV(3,3),IDENT(3,3)
      COMMON /MATRB/ S(3,3),CA(3,3),CB(3,3),CE(3,3),CF(3,3)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
C
      IF (IABORT.EQ.1) RETURN
C
      POISM = POIS(3)
      KMAT = K(3)
      GMAT = G(3)
C
      KTEMP = K(1)
      GTEMP = G(1)
      DO 30 INCL = 1,2
         IF (INCL.EQ.2.AND.IKIND.EQ.0) THEN
            GTEMP = 0.0
            KTEMP = 0.0
```

```
      ELSE
         GTEMP = G(INCL)
         KTEMP = K(INCL)
      ENDIF
C
      ALPHA = 2.0*(5.0*POISM-1)+10.0*(1-POISM)*(KMAT/(KTEMP-KMAT)-
     *    GMAT/(GTEMP-GMAT))
      BETA(INCL) = 2.0*(4.0-5.0*POISM)+15.0*(1-POISM)*(GMAT/(GTEMP-
     *    GMAT))
      ZETA1 = 12.0*(13.0*POISM-14.0*POISM**2)-(96.0*ALPHA/(3.0*ALPHA+
     *    2.0*BETA(INCL)))*(1-2.0*POISM)*(1+POISM)
      ZETA2 = 6.0*(25.0-34.0*POISM+22.0*POISM**2)-(36.0*ALPHA/(3.0*
     *    ALPHA+2.0*BETA(INCL)))*(1-2.0*POISM)*(1+POISM)
C
      DO 20 I = 1,3
         DO 10 J = 1,3
            IF (INCL.EQ.1.AND.IMORI.NE.0) THEN
               WI(I,J) = ZETA1
               IF (I.EQ.J) WI(I,J) = ZETA1+2*ZETA2
            ELSEIF (INCL.EQ.1.AND.IMORI.EQ.0) THEN
               WI(I,J) = 0.0
            ELSEIF (INCL.EQ.2.AND.IMORI.EQ.2) THEN
               WV(I,J) = ZETA1
               IF (I.EQ.J) WV(I,J) = ZETA1+2*ZETA2
            ELSE
               WV(I,J) = 0.0
            ENDIF
   10       CONTINUE
   20    CONTINUE
   30 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE CALCS(IABORT)
C==== calculate Eshelby matrices SI and SV
      REAL IDENT,K,KCMP,MAG
      COMMON /MATRB/ S(3,3),CA(3,3),CB(3,3),CE(3,3),CF(3,3)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
C
      IF (IABORT.EQ.1) RETURN
C
      POISM = POIS(3)
      SDET = 15.0*(1-POISM)
C
      S1 = 5.0*POISM-1
      S2 = 4.0-5.0*POISM
C
      DO 20 I = 1,3
         DO 10 J = 1,3
            S(I,J) = S1/SDET
            IF (I.EQ.J) S(I,J) = (S1+2.0*S2)/SDET
   10    CONTINUE
   20 CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE GAMMA(A,CONC,ITYPE,YMULT,IABORT)
C==== calculate correction matrix A given inclusion I and its
C     concentration CONC, Y depends on microstructural features
C     ITYPE identifies inclusion or void
      REAL IDENT,K,KCMP,MAG
      COMMON /MATRA/ BETA(2),WI(3,3),WV(3,3),IDENT(3,3)
      DIMENSION A(3,3)
C
      IF (IABORT.EQ.1) RETURN
C
      Y = YMULT*(1-CONC)/24.0
      MAG = 5.0*CONC*Y/(4.0*BETA(ITYPE)**2)
```

```
      DO 20 I = 1,3
         DO 10 J = 1,3
            IF (ITYPE.EQ.1) A(I,J) = IDENT(I,J)+MAG*WI(I,J)
            IF (ITYPE.EQ.2) A(I,J) = IDENT(I,J)+MAG*WV(I,J)
   10    CONTINUE
   20 CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE ADD(C,A,B)
C==== subroutine for adding two square matrices C=A+B
      DIMENSION A(3,3),B(3,3),C(3,3)
C
      DO 20 I = 1,3
         DO 10 J = 1,3
            C(I,J) = A(I,J)+B(I,J)
   10    CONTINUE
   20 CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE SUB(C,A,B)
C==== subroutine for adding two square matrices C=A-B
      DIMENSION A(3,3),B(3,3),C(3,3)
C
      DO 20 I = 1,3
         DO 10 J = 1,3
            C(I,J) = A(I,J)-B(I,J)
   10    CONTINUE
   20 CONTINUE
C
      RETURN
      END
```

```
C
C
      SUBROUTINE MULT(C,A,B)
C==== subroutine for multiplying two square matrices C=A.B
      DIMENSION A(3,3),B(3,3),C(3,3)
C
      DO 30 I = 1,3
         DO 20 J = 1,3
            C(I,J) = 0
            DO 10 K = 1,3
               C(I,J) = C(I,J)+A(I,K)*B(K,J)
   10       CONTINUE
   20    CONTINUE
   30 CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE INVERT(AI,A,IABORT)
C==== subroutine used for inverting matrix A to give AI
      DIMENSION A(3,3),AI(3,3)
C
      IF (IABORT.EQ.1) RETURN
C
      DETA = -(A(1,3)*A(2,2)*A(3,1))+A(1,2)*A(2,3)*A(3,1)+A(1,3)*
     *   A(2,1)*A(3,2)-A(1,1)*A(2,3)*A(3,2)-A(1,2)*A(2,1)*A(3,3)+
     *   A(1,1)*A(2,2)*A(3,3)
C
      IF (DETA.NE.0) THEN
         AI(1,1) = (-(A(2,3)*A(3,2))+A(2,2)*A(3,3))/DETA
         AI(1,2) = (A(1,3)*A(3,2)-A(1,2)*A(3,3))/DETA
         AI(1,3) = (-(A(1,3)*A(2,2))+A(1,2)*A(2,3))/DETA
         AI(2,1) = (A(2,3)*A(3,1)-A(2,1)*A(3,3))/DETA
         AI(2,2) = (-(A(1,3)*A(3,1))+A(1,1)*A(3,3))/DETA
         AI(2,3) = (A(1,3)*A(2,1)-A(1,1)*A(2,3))/DETA
         AI(3,1) = (-(A(2,2)*A(3,1))+A(2,1)*A(3,2))/DETA
```

```
         AI(3,2) = (A(1,2)*A(3,1)-A(1,1)*A(3,2))/DETA
         AI(3,3) = (-(A(1,2)*A(2,1))+A(1,1)*A(2,2))/DETA
      ELSE
         IABORT = 1
         WRITE (6,'(A)') ' SBR INVERT: indeterminant matrix'
      ENDIF
C
      RETURN
      END
C
C
      SUBROUTINE GAUWRT(NDIST,NPTS,IABORT)
C==== write out cumulative distribution data.
C     for some reason, cannot print out PROBs correctly using
C     F format, numbers end up getting multiplied by ten.
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
C
      IF (IABORT.EQ.1) RETURN
      WRITE (6,'(A)') ' Writing GAUSS.DAT'
      OPEN (UNIT=7,FILE='_GAUSS.DAT',FORM='FORMATTED',STATUS='UNKNOWN')
      WRITE (7,5000)
      DO 10 IPTS = 1,NPTS
         WRITE (7,5100) Z(IPTS),(PROB(IDIST,IPTS),RADIUS(IDIST,IPTS),
     *      IDIST = 1,NDIST)
   10 CONTINUE
      CLOSE (7)
C
      RETURN
 5000 FORMAT ('    Z          Pr          Radius(mm)          Pr
     *    Radius(mm)          Pr          Radius(mm)')
 5100 FORMAT (1X,F6.3,6(3X,0PE13.6))
```

```
      END
C
C
      SUBROUTINE HSTWRT(NDIST,NPTS,IABORT)
C==== write out histogram and tracking data
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
C
      IF (IABORT.EQ.1) RETURN
      WRITE (6,'(A)') ' Writing HISTO.DAT'
      OPEN (UNIT=7,FILE='_HISTO.DAT',FORM='FORMATTED',STATUS='UNKNOWN')
      NPRTOT = 0
      VOLTOT = 0
      DO 20 IDIST = 1,NDIST
         DO 10 IHST = 1,NPTS
            NPRTOT = NPRTOT+NUMPAR(IDIST,IHST)
            VOLTOT = VOLTOT+VOLPAR(IDIST,IHST)
   10    CONTINUE
   20 CONTINUE
C
      CUMVOL = 0.0
      DO 40 IDIST = 1,NDIST
         WRITE (7,5000)
         DO 30 IHST = 1,NPTS
            PERNPR = 100*REAL(NUMPAR(IDIST,IHST))/REAL(NPRTOT)
            PERVOL = 100*VOLPAR(IDIST,IHST)/VOLTOT
            CUMVOL = CUMVOL+PERVOL
            WRITE (7,5100) IHST,RADIUS(IDIST,IHST),
     *         ALOG10(NUMPAR(IDIST,IHST)),VOLPAR(IDIST,IHST),PERNPR,
     *         PERVOL,PROB(IDIST,IHST),CUMVOL
   30    CONTINUE
   40 CONTINUE
```

```
      CLOSE (7)
C
      RETURN
 5000 FORMAT (' Point    avg R(mm)     log # part.    volume(mm3)  %no.p
     *articles   % part.volume   cum. prob.      cum. vol.')
 5100 FORMAT (2X,I3,3X,7(1PE13.6,2X))
      END
C
C
      SUBROUTINE STRWRT(NDIST,NPTS,VOLSMP,GAMM,FDBND,YMULT,IKIND,IMORI,
     *   IPOIS,PRESS,DILATO,FILNM,NLTOT,STSFUD,IWRT,DBANG,ISKIP,IABORT)
C==== write out true stress and dilatation results versus critical strain
C     include probability survival, radius, no. particles and
C     distribution info. write intermediate data where e_i<e_max if
C     IWRT>0.
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AKO(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      CHARACTER FILNM*8
C
      IF (IABORT.EQ.1) RETURN
C
      IF (FILNM.EQ.'DEFAULT') FILNM = '_STRESS'
      WRITE (6,'(/,A,A8,A)') ' Writing to ',FILNM,'.DAT'
```

```
C
      OPEN (UNIT=7,FILE=FILNM//'.DAT',STATUS='UNKNOWN')
      WRITE (7,5000)
      DO 10 I = 1,NDIST
         WRITE (7,'(1X,I1,4(3X,0PE11.4))') I,RADAVG(I),LOGSTD(I),
     *      VLFRFO(I),VLFRVO(I)
   10 CONTINUE
C
      GCTOT = 0
      WRITE (7,5100) G(3),G(1),POIS(3),POIS(1),G(2),K(2)
      WRITE (7,5200) VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS
      WRITE (7,5300) PRESS,GAMM,DILATO,STSFUD,DBANG
      WRITE(7,5600) AKO(1),AK1(1),AK2(1),ISKIP
      WRITE (7,5400)
      DO 20 I = 1,NDIST*NPTS+NLTOT+1
         ETMP = ECMP(I)/1E6
         IF(NETVV(I).GT.0)THEN
            GCDADC = -2.0*GAMM*DADC(I)/(VOLSMP*1E-3)
            GCTOT = GCTOT+GCDADC/1E6
         ENDIF
         IF (IWRT.EQ.0.AND.IPDIST(I).LT.9.AND.I.GT.NLTOT+1) GOTO 20
         WRITE (7,5500) I,CRTSTN(I),STRESS(I),DILAT(I),PRBSRV(I),ETMP,
     *      POISC(I),STRNP1(I),GCTOT,NETVV(I),IPDIST(I)
   20 CONTINUE
C
      CLOSE (7)
      RETURN
 5000 FORMAT (' #    avg Rad(um)    std dev          Vf           Vv')
 5100 FORMAT (' Gm(Pa)=',0PE11.4,' Gf(Pa)=',0PE11.4,' vm=',0PE11.4,
     *   ' vf=',0PE11.4,' Gv(Pa)=',0PE11.4,' Kv=',0PE11.4)
 5200 FORMAT (' V(mm3) =',0PE11.4,' frac dbnd=',0PE11.4,' Y-mult=',0PE11
     *   .4,' w-type=',I3,' m-type=',I3,' v-type=',I3)
 5300 FORMAT (' PO(Pa)=',0PE11.4,' Gc(Pa-m)=',0PE11.4,' (dV/V)O=',0PE11
     *   .4, ' crit.strs(Pa)=',0PE11.4, ' dbnd.ang(deg)=',0PE11.4)
 5400 FORMAT (' Point  crit strn    stress(MPa)     dV/V       Pr|surv
     *    E_c(MPa)     Poisson      mat strn    CumGAC(MPa)      V_v
     *  dist')
```

```
 5600 FORMAT (' AK0(Pa)=',0PE11.4,' AK1(Pa)=',0PE11.4,' AK2(Pa)=',
     *   0PE11.4,' ISKIP= ',I2)
 5500 FORMAT (1X,I3,3X,9(1PE11.4,2X),1X,I1)
C
      END
C
C
      SUBROUTINE STRAUX(NDIST,NPTS,VOLSMP,GAMM,FDBND,YMULT,IKIND,IMORI,
     *   IPOIS,PRESS,DILATO,FILNM,NLTOT,STSFUD,IWRT,DBANG,ISKIP,IABORT)
C==== write out true stress and dilatation results versus critical strain
C     include probability survival, radius, no. particles and
C     distribution info. write intermediate data where e_i<e_max if
C     IWRT>0.
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /DIST/ RADAVG(NTDIS),LOGSTD(NTDIS),VLFRFO(NTDIS),
     *   VLFRVO(NTDIS)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /PROPA/ K(3),G(3),E(3),POIS(3),CI(3,3),CV(3,3),CO(3,3)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /PROPC/ AK0(1),AK1(1),AK2(1),STRNP1(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
      CHARACTER FILNM*8
C
      IF (IABORT.EQ.1) RETURN
C
      IF (FILNM.EQ.'DEFAULT') FILNM = '_STRESS'
      WRITE (6,'(/,A,A8,A)') ' Writing to ',FILNM,'.AUX'
C
      OPEN (UNIT=7,FILE=FILNM//'.AUX',STATUS='UNKNOWN')
```

```
      WRITE (7,5000)
      DO 10 I = 1,NDIST
         WRITE (7,'(1X,I1,4(3X,OPE11.4))') I,RADAVG(I),LOGSTD(I),
     *      VLFRFO(I),VLFRVO(I)
   10 CONTINUE
C
      GCTOT = 0
      WRITE (7,5100) G(3),G(1),POIS(3),POIS(1),G(2),K(2)
      WRITE (7,5200) VOLSMP,FDBND,YMULT,IKIND,IMORI,IPOIS
      WRITE (7,5300) PRESS,GAMM,DILATO,STSFUD,DBANG
      WRITE(7,5600) AKO(1),AK1(1),AK2(1),ISKIP
      WRITE (7,5400)
      DO 20 I = 1,NDIST*NPTS+NLTOT+1
         ETMP = ECMP(I)/1E6
         IF(NETVV(I).GT.0)THEN
            DCV = NETVV(I)-NETVV(I-1)
            GCDADC = (-2.0*GAMM*DADC(I)/(VOLSMP*1E-3))/1E6
            GCTOT = GCTOT+GCDADC
         ENDIF
         IF (IWRT.EQ.0.AND.IPDIST(I).LT.9.AND.I.GT.NLTOT+1) GOTO 20
         WRITE (7,5500) I,CRTSTN(I),STRESS(I),DILAT(I),PRBSRV(I),ETMP,
     *      GCDADC,DCV,GCTOT,NETVV(I),IPDIST(I)
   20 CONTINUE
C
      CLOSE (7)
      RETURN
 5000 FORMAT (' #    avg Rad(um)    std dev          Vf           Vv')
 5100 FORMAT (' Gm(Pa)=',OPE11.4,' Gf(Pa)=',OPE11.4,' vm=',OPE11.4,
     *   ' vf=',OPE11.4,' Gv(Pa)=',OPE11.4,' Kv=',OPE11.4)
 5200 FORMAT (' V(mm3) =',OPE11.4,' frac dbnd=',OPE11.4,' Y-mult=',OPE11
     *   .4,' w-type=',I3,' m-type=',I3,' v-type=',I3)
 5300 FORMAT (' PO(Pa)=',OPE11.4,' Gc(Pa-m)=',OPE11.4,' (dV/V)O=',OPE11
     *   .4, ' crit.strs(Pa)=',OPE11.4, ' dbnd.ang(deg)=',OPE11.4)
 5400 FORMAT (' Point  crit strn    stress(MPa)     dV/V       Pr|surv
     *    E_c(MPa)   GCDADC(MPa)     dC_cal     CumGAC(MPa)      V_v
     *  dist')
 5600 FORMAT (' AKO(Pa)=',OPE11.4,' AK1(Pa)=',OPE11.4,' AK2(Pa)=',
```

```
     *   OPE11.4,' ISKIP= ',I2)
 5500 FORMAT (1X,I3,3X,9(1PE11.4,2X),1X,I1)
C
      END
C
C
      SUBROUTINE DBGWRT(NDIST,NPTS,NLTOT,IABORT)
C==== write out additional data for debugging purposes.
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
C
      IF (IABORT.EQ.1) RETURN
      WRITE (6,'(A)') ' Writing DEBUG.DAT'
C
      OPEN (UNIT=7,FILE='_DEBUG.DAT',FORM='FORMATTED',STATUS='UNKNOWN')
      WRITE (7,5000)
      DO 10 IHST = 1,NDIST*NPTS+NLTOT+1
         WRITE (7,5100) IHST,NETVF(IHST),NETVV(IHST),DADC(IHST),
     *      C11(IHST),C12(IHST),C21(IHST),C22(IHST),C23(IHST),
     *      PRBSRV(IHST)
   10 CONTINUE
      CLOSE (7)
C
      RETURN
```

```
 5000 FORMAT (' Point     net Vf          net Vv          dA/dc
     * C11            C12            C21            C22            C23
     *     Pr|surv')
 5100 FORMAT (2X,I3,3X,9(0PE13.6,2X))
      END
C
C
      SUBROUTINE DBGRAT(NDIST,NPTS,NLTOT,IABORT)
C==== write out additional data for debugging purposes
C     along with true stress-strain data outputs radius and the factor
C     SQRT(RAD*dG/dc) to look at its relationship with crit. strain
C
C     set NPTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      REAL IDENT,K,KCMP,KTMP,MAG
      INTEGER GSMX,NPTMX
      PARAMETER (GSMX = 1000,NPTMX = 1000,NTDIS = 1)
      COMMON /GAUS/ Z(GSMX),RADIUS(NTDIS,GSMX),PROB(NTDIS,GSMX)
      COMMON /DEBUG/ NUMPAR(NTDIS,GSMX),VOLPAR(NTDIS,GSMX),
     *   NETVF(NPTMX),NETVV(NPTMX),DADC(NPTMX),NPARTL(NTDIS)
      COMMON /PROPB/ C11(NPTMX),C12(NPTMX),C21(NPTMX),C22(NPTMX),
     *   C23(NPTMX),ECMP(NPTMX),POISC(NPTMX)
      COMMON /RESULT/ CRTSTN(NPTMX),STRESS(NPTMX),DILAT(NPTMX),
     *   PRBSRV(NPTMX),SORRAD(NPTMX),SORPAR(NPTMX),SORVLP(NPTMX),
     *   IPDIST(NPTMX)
C
      IF (IABORT.EQ.1) RETURN
      WRITE (6,'(A)') ' Writing DERAT.DAT'
C
      OPEN (UNIT=7,FILE='_DERAT.DAT',FORM='FORMATTED',STATUS='UNKNOWN')
      WRITE (7,5000)
      DO 10 IHST = 1,NDIST*NPTS+NLTOT+1
         ETMP = ECMP(IHST)
         RAD = 0.0
         DNETF = 0.0
         DNETV = 0.0
```

```
          DETMP = 0.0
          IF (IHST.GT.NLTOT+1) RAD = SORRAD(IHST-NLTOT-1)
          IF (IHST.GT.1) DNETF = ABS(NETVF(IHST)-NETVF(IHST-1))
          IF (IHST.GT.1) DNETV = ABS(NETVV(IHST)-NETVV(IHST-1))
          IF (IHST.GT.1) DETMP = ABS(ECMP(IHST)-ECMP(IHST-1))
          FACT = SQRT(RAD*DETMP)
          WRITE (7,5100) IHST,CRTSTN(IHST),STRESS(IHST),RAD,
     *       PRBSRV(IHST),ETMP,POISC(IHST),DNETF,DNETV,FACT
   10 CONTINUE
      CLOSE (7)
C
      RETURN
 5000 FORMAT (' Point  crit strn    stress(MPa)  Avg r(mm)      Pr|surv
     *    E_c(MPa)     Poisson       dV_f          dV_v          fact')
 5100 FORMAT (1X,I3,3X,9(1PE11.4,2X))
      END
```

```
      INCLUDE 'MSGRAPH.FOR'
C
      SUBROUTINE CRVPLT(NDIST,NPTS,NLTOT,IWRT,IABORT)
C==== driver routine for plotting curve on screen, keep the
C     INCLUDE 'MSGRAPH.FOR' with this module.
C
C     set PTMX = NTDIS*GSMX
C
      REAL LOGSTD,NPARTL,NUMPAR,NETVF,NETVV
      INTEGER GSMX,PTMX
      PARAMETER (GSMX = 1000,PTMX = 1000,NTDIS = 1)
      COMMON /RESULT/ CRTSTN(PTMX),STRESS(PTMX),DILAT(PTMX),
     *   PRBSRV(PTMX),SORRAD(PTMX),SORPAR(PTMX),SORVLP(PTMX),
     *   IPDIST(PTMX)
      DIMENSION X(PTMX),Y1(PTMX),Y2(PTMX)
      CHARACTER ANS*1
C
      IF (IABORT.EQ.1) RETURN
C
   10 CONTINUE
      WRITE (6,'(/,A)') ' Graph results on screen? (Y/N)'
      READ (5,'(A1)') ANS
C
      NTOT=0
      DO 20 I = 1,NDIST*NPTS+NLTOT+1
         IF (IWRT.EQ.0.AND.IPDIST(I).LT.9.AND.I.GT.NLTOT+1) GOTO 20
         NTOT = NTOT+1
         X(NTOT) = CRTSTN(I)
         Y1(NTOT) = STRESS(I)
         Y2(NTOT) = DILAT(I)
   20 CONTINUE
C
      IF (ANS.EQ.'Y') THEN
         WRITE (6,'(A)') ' Strain, Stress and dV/V end pts'
         READ (5,*) XEND,YSEND,YDEND
         CALL GRAF(NTOT,X,Y1,Y2,XEND,YSEND,YDEND)
      ELSE
```

```
      ENDIF
C
      IF (ANS.EQ.'Y') GOTO 10
C
      RETURN
      END
C
C
      SUBROUTINE GRAF(N,X,Y1,Y2,XEND,YSEND,YDEND)
C
C     set PTMX = NTDIS*GSMX
C
      INTEGER PTMX
      PARAMETER (GSMX = 1000,PTMX = 1000,NTDIS = 1)
      DIMENSION X(PTMX),Y1(PTMX),Y2(PTMX)
C
      INCLUDE 'GRFDEF.FOR'
C
      CALL VIDEO(MAXX,MAXY,NOGRAF)
      IF (NOGRAF.EQ.0) THEN
C
         CALL VWPORT(MAXX,MAXY)
C
         XBEG = 0
         YBEG = 0
         CALL WINDOW(XBEG,YBEG,XEND,YSEND)
         ICURV = 1
         XLAB = 'strain'
         YLAB = 'strs (MPa)'
         CALL ATTRIB(ICURV,ILNCOL,ILNSTY)
         CALL LABELS(ICURV,ILNCOL,XLAB,YLAB,XBEG,YBEG,XEND,YSEND)
C
         ICURV = 1
         CALL ATTRIB(ICURV,ILNCOL,ILNSTY)
         CALL CURVE(X,Y1,N,ILNCOL,ILNSTY)
C
         CALL WINDOW(XBEG,YBEG,XEND,YDEND)
```

```
         ICURV = 3
         XLAB = 'strain'
         YLAB = 'dV/V'
         CALL ATTRIB(ICURV,ILNCOL,ILNSTY)
         CALL LABELS(ICURV,ILNCOL,XLAB,YLAB,XBEG,YBEG,XEND,YDEND)
C
         ICURV = 3
         CALL ATTRIB(ICURV,ILNCOL,ILNSTY)
         CALL CURVE(X,Y2,N,ILNCOL,ILNSTY)
C
         CALL ENDGRF()
C
      ELSE
         WRITE (6,'(A)') ' SBR GRAF: problem with graphics'
      ENDIF
C
      RETURN
      END
```

INTERNAL DISTRIBUTION

DREV R-9815

1 – Deputy Director General
1 – Chief Scientist
1 – H/EM Section
6 – Document Library
1 – F.C. Wong (Author)
1 – F. Beaupre
1 – Dr. S. Thiboutot
1 – S. Villeneuve
1 – Dr. D. Nandlall
1 – Dr. G. McIntosh

EXTERNAL DISTRIBUTION

DREV R–9815

1 – Aeronautical and Maritime Research Laboratory
Explosives Ordnance Division
Salisbury, SA 5108
Attn: Dr. Sook Ying Ho

2 – Defence Research Agency
Propulsion Research WX-4
Fort Halstead
Sevenoaks, Kent TN 14 7BP
United Kingdom
Attn: Dr. Jim Buswell

1 – Dept. of Materials Engineering
Composite Group
The University of British Columbia
23324 Main Mall
Vancouver, B.C.
V6T 1Z4
Attn: Dr. A. Poursarstip

EXTERNAL DISTRIBUTION

DREV R–9815

1 – DRDIM
1 – DRDIM (unbound copy)
1 – CRAD
1 – DSAL
1 – DSAA
1 – DSAM
1 – DAEPM(FT) 3
1 – DLR
1 – DMCS
1 – DARFT
1 – DNR

1 – Bristol Aerospace Ltd.
Rockets and Space Division
P.O. Box 874
Winnipeg, MB
R3C 2S4
Attn: Engineering Library

1 – National Research Council Canada
National Aeronautical Establishment
Structures and Materials Laboratory
Ottawa, Ontario
Attn: Dr. W. Wallace

1 – U. S. Army Missle Command
AMSMI-RD-PR-M (B. Marsh)
Redstone Arsenal, AL 35898-5249
United States of America
Attn: Commander

DOCUMENT CONTROL DATA

1. ORIGINATOR (name and address)	2. SECURITY CLASSIFICATION (Including special warning terms if applicable)
Defence Research Establishment Valcartier 2459 Pie-XI Blvd. North Val-Bélair, Qc G3J 1X5	UNCLASSIFIED

3. TITLE (Its classification should be indicated by the appropriate abbreviation (S, C, R or U)
Analysis of Particulate Composite Behavior Based on Nonlinear Elasticity and an Improved Mori-Tanaka Theory

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)
Wong, Franklin C.

5. DATE OF PUBLICATION (month and year)	6a. NO. OF PAGES	6b. NO. OF REFERENCES
September 1998	70	76

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. Give the inclusive dates when a specific reporting period is covered.)
DREV Report

8. SPONSORING ACTIVITY (name and address)

9a. PROJECT OR GRANT NO. (Please specify whether project or grant)	9b. CONTRACT NO.
WU 3ee15 Service Life Prediction Methodologies	

10a. ORIGINATOR'S DOCUMENT NUMBER	10b. OTHER DOCUMENT NOS
DREV R-9815	N/A

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

- [x] Unlimited distribution
- [ ] Contractors in approved countries (specify)
- [ ] Canadian contractors (with need-to-know)
- [ ] Government (with need-to-know)
- [ ] Defense departments
- [ ] Other (please specify)

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

A micromechanical model for the analysis of particulate mechanical behavior is presented. Nonlinear effects are introduced in the model by a nonlinear elastic description of the matrix and through a modulus degradation routine. The first part of the study uses the experimental data from a range of glass bead/HTPB composites to back calculate the model parameters. The results showed that the model gave a good representation of the processes believed to control mechanical behavior. These processes include partial particle debonding and progressive debonding from the largest to smallest particles throughout the strain history. The second part of the study examines the sensitivity of the model results to small changes in the adjustable input parameters. The residual bond in a debonded particle was found to have a dominating effect on the calculated results. Based on the sensitivity results, "best guess" interaction and debonding parameters were selected to examine the predictive capability of the model. For glass bead/HTPB composites, the predicted composite stresses were within 10% of the experimental data. Dilatation was usually over-predicted. For glass bead/polyethylene and glass bead/polyurethane data found in the literature, predicted composite stresses were within 15% to 24%, respectively. The results showed that the model was capable of predicting the mechanical behavior of composites comprised of glass beads in HTPB, PU or HDPE matrices as long as characteristic adhesive parameters were available for each system.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

micromechanics, Mori-Tanaka, particle interaction, debond, 2-phase, 3 phase, void, vacuole, Eshelby, elastic properties,
prediction, energy balance, mechanical behavior, eigenstrain, nonlinear